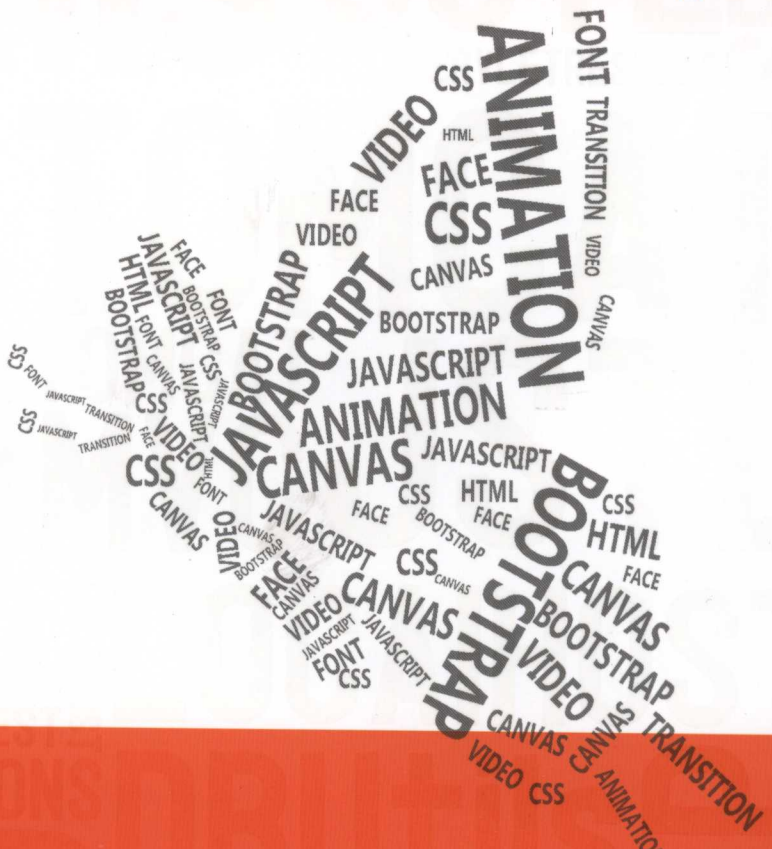
工业和信息化“十三五”
人才培养规划教材

国家信息技术紧缺人才培养工程
指定教材

响应式 Web 开发项目教程 (HTML5+CSS3+Bootstrap)

黑马程序员 / 编著



本书涵盖了响应式 Web 开发领域中 HTML5、CSS3 和 Bootstrap 框架的操作技术和应用设计规范，每个知识点都应用到了实战项目中，共 18 个经典项目。

提供免费教学资源，包括 8 个精美教学 PPT、13 个教材补充项目、1000 道测试题、长达 40 小时的教学视频等。

添加 QQ 或微信号：208695827，获取教学答案、源码和“助学金红包”。

中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

播妞

播妞—— IT技术女神，由传智播客旗下高端教育品牌黑马程序员推出，专门服务于计算机相关专业的大学生及IT爱好者；可随时提供教材源代码、习题答案、免费视频教程和就业宝典等。

播妞倾情寄语：

你加，或者不加我
我就在这里
不离 不弃
来我这里
或者
让我住进你的心里

★ 小心！此处常有“助学金红包”出没！★
快来领取！

播妞QQ：208695827

播妞微信：208695827

教师获取教材配套资源

添加微信/QQ

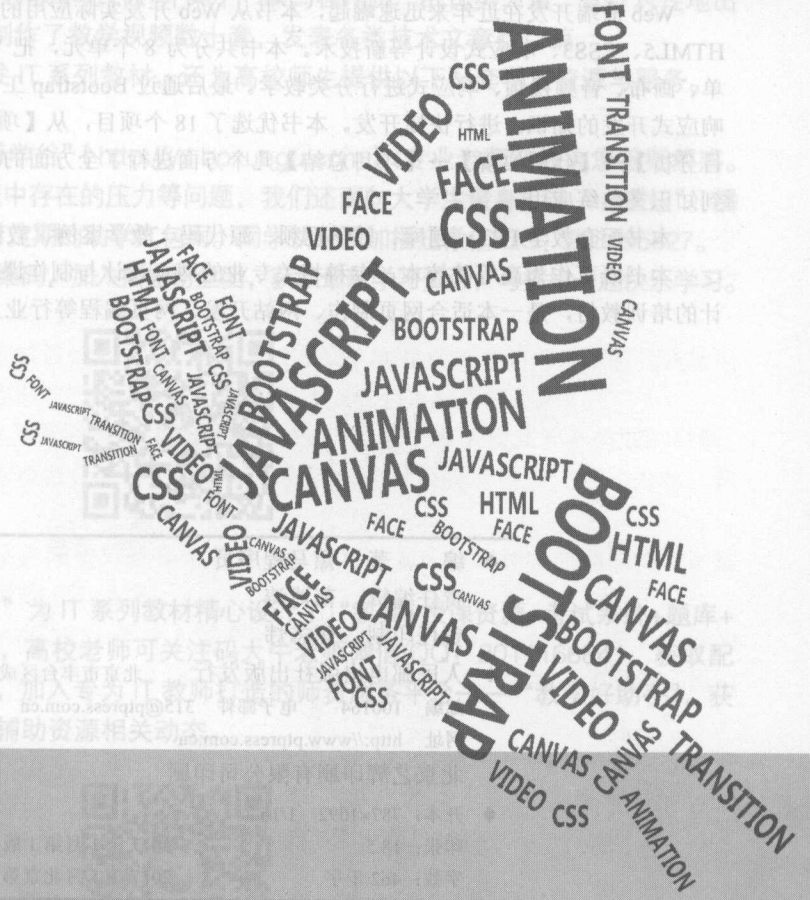
2011168841

工业和信息化“十三五”
人才培养规划教材

NITE 国家信息技术紧缺人才培养工程
指定教材

响应式Web开发项目教程
(HTML5+CSS3+Bootstrap)

黑马程序员 / 编著



人 民 邮 电 出 版 社
北 京

图书在版编目（CIP）数据

响应式Web开发项目教程 : HTML5+CSS3+Bootstrap / 黑马程序员编著. -- 北京 : 人民邮电出版社, 2017.1
工业和信息化“十三五”人才培养规划教材
ISBN 978-7-115-43934-5

Ⅰ. ①响… Ⅱ. ①黑… Ⅲ. ①超文本标记语言－程序设计－高等学校－教材②网页制作工具－高等学校－教材
Ⅳ. ①TP312②TP393.092

中国版本图书馆CIP数据核字(2016)第301271号

内 容 提 要

Web 前端开发在近年来迅速崛起，本书从 Web 开发实际应用的角度，以项目式的教学方式讲解 HTML5、CSS3、响应式设计等新技术。本书共分为 8 个单元，把 Web 开发项目按文本、图文、表单、画布、音频视频、响应式进行分类教学。最后通过 Bootstrap 工具开发来整合 HTML5、CSS3 和响应式开发的知识，进行快捷开发。本书优选了 18 个项目，从【项目描述】→【前导知识】→【项目分析】→【代码实现】→【项目总结】几个方面进行了全方面的讲解，让读者可以真正做到既学到知识又熟练应用。

本书配套教学 PPT、题库、教学视频、源代码、教学案例、教学设计等资源。

本书既可作为高等院校本、专科相关专业的网页设计与制作课程的教材，也可作为网页平面设计的培训教材，是一本适合网页制作、网站开发、网页编程等行业人员阅读与参考的读物。

◆ 编　　著　黑马程序员
责任编辑　范博涛
责任印制　焦志炜

◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路 11 号
邮编　100164　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京艺辉印刷有限公司印刷

◆ 开本：787×1092　1/16
印张：18.5　　2017 年 1 月第 1 版
字数：462 千字　　2017 年 1 月北京第 1 次印刷

定价：42.00 元

读者服务热线：(010)81055256　印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东工商广字第 8052 号

序 言 PREFACE

江苏传智播客教育科技股份有限公司（简称传智播客）是一家致力于培养高素质软件开发人才的科技公司，“黑马程序员”是传智播客旗下高端 IT 教育品牌。

“黑马程序员”的学员多为大学毕业后，想从事 IT 行业，但各方面条件还不成熟的年轻人。“黑马程序员”的学员筛选制度，非常严格，筛选制度包括了严格的技术测试、自学能力测试，还包括性格测试、压力测试、品德测试等。百里挑一的残酷筛选制度确保了学员质量，并降低了企业的用人风险。

自“黑马程序员”成立以来，教学研发团队一直致力于打造精品课程资源，不断在产、学、研 3 个层面创新自己的执教理念与教学方针，并集中“黑马程序员”的优势力量，有针对性地出版了计算机系列教材 30 多册，制作了教学视频数十套，发表各类技术文章数百篇。

“黑马程序员”不仅斥资研发 IT 系列教材，还为高校师生提供以下配套学习资源与服务。

为大学生提供的配套服务：

1. 专业的辅助学习平台“博学谷”（http://yx.boxuegu.com），专业老师在线为您答疑解惑。
2. 针对高校学生在学习过程中存在的压力等问题，我们还面向大学生量身打造了“播妞”。播妞不仅致力推行快乐学习，还会有定期的助学红包雨。同学快来添加播妞微信/QQ：208695827。
3. 高校学生也可扫描右方二维码，加入播妞粉丝团，获取最新学习资源，与播妞一起快乐学习。

为 IT 教师提供的配套服务：

针对高校教学，“黑马程序员”为 IT 系列教材精心设计了“教案+授课资源+考试系统+题库+教学辅助案例”的系列教学资源，高校老师可关注码大牛老师微信/QQ：2011168841，获取配套资源，也可以扫描右方二维码，加入专为 IT 教师打造的师资服务平台——“教学好助手”，获取“黑马程序员”最新教师教学辅助资源相关动态。

传智播客和黑马程序员

2016年9月

前言 FOREWORD

随着互联网行业的持续发展，移动互联网等新业务不断发展壮大，海量的平台开发工作形成了巨大的人才缺口，尤其是 Web 前端、移动端 HTML5 开发人才更为紧缺。随着互联网业内竞争的不断加剧，企业平台开始在界面友好性和操作方便性方面投入更多的精力，只有平台更加人性化，才能吸引更多的用户，所以 Web 开发人员的地位在业内开始迅速提高。

为什么要学习本书

一个优秀的 Web 开发工程师需要具备一定的综合素质才能胜任企业日益复杂多变的工作要求。这些素质包括熟知页面布局，熟练样式美化，掌握 JavaScript 基础，熟悉 Bootstrap 响应式布局设计，能够使用 HTML5+CSS3 开发出炫丽的移动端交互效果等，而本书正为此应运而生。

如何使用本书

本书分为 8 个单元，每个单元由 2~4 个项目组成，以项目为导向串讲 Web 开发中的知识点。为什么我们摒弃了传统的知识点讲解方法，就是为了能够培养学生的编程兴趣，达到自主学习的目的。我们发现人们玩游戏的兴趣远比去学习一堆枯燥的代码要大得多，这是因为玩游戏是一个自我挑战的过程，一个好玩的游戏有精准的难度设计、炫酷的视觉感受，恰当好处地勾起你挑战下一关的欲望。所以我们在想，为什么不可以像设计游戏一样去设计一本书。

单元 1，先告诉你什么是响应式设计以及 HTML5 和 CSS3。本章要求很简单，照着代码敲出两个项目，看 HTML5 和 CSS3 能够做出什么。相信你一定可以做到，并且感兴趣。

单元 2，主打文字主题，在这里需要挑战 3 个项目——电子杂志、唯美软文和手机版的导航。每个挑战项目都会有“指南”，如项目介绍、前导知识，并且每个项目都有很多注释的内容，帮助读者解读关键性代码。

单元 3，主打图文主题，这里也需要挑战 3 个项目——书城、多肉植物和会晃动的桃子，技术上包括了 CSS3 的阴影、圆角和动画效果等。

单元 4，表单。它可以说是 HTML5 最令人振奋的改进之一，帮助开发人员省去了大段的 JavaScript 验证代码。通过第 4 单元的两个项目，你可以充分地感受到 HTML5 带给前端开发人员的便捷。

单元 5，画布。它是 HTML5 新加的元素。本单元有两个特别好玩的项目——网页涂鸦板和红包照片。

单元 6，音频和视频。如果读者是一个爱听音乐爱看电影的年轻人。那么在这个单元，可以定制一个属于自己的视频播放器和音频播放器。

单元 7，我们将迎来里程碑式的内容——响应式 Web 设计原理。本单元会有 2 个项目，带领读者第一次完成响应式项目。看着自己制作的网页会在各种设备的屏幕上任意切换，一定很有成就感。

单元 8，Bootstrap 响应式 Web 开发实战。Bootstrap 是响应式 Web 开发神器，本单元通过 1 个完整的项目，让你快速地搭建响应式网站。

为了保证读者学习的全面性，我们在全书的最后制作了丰富的附录，内容包括 HTML5 新增标签、HTML5 废除标签、CSS3 新增属性和需要加浏览器私有前缀的属性等。

致谢

本教材的编写和整理工作由传智播客教育科技股份有限公司完成，主要参与人员有吕春林、马丹、金鑫、王玉华、马伦、刘晓强、赵玉川、周淑刚、汪磊等，全体人员在这近一年的编写过程中付出了很多辛勤的汗水，在此一并表示衷心的感谢。

意见反馈

尽管我们尽了最大的努力，但教材中难免会有不妥之处，欢迎各界专家和读者朋友们来信来函给予宝贵意见，我们将不胜感激。您在阅读本书时，如发现任何问题或有不认同之处可以通过电子邮件与我们取得联系。

请发送电子邮件至 itcast_book@vip.sina.com。

黑马程序员

2016-9-8 于北京

目 录 CONTENTS

Responsive Web Design

1 Chapter

单元 1
响应式和 HTML5+CSS3 初体验

十几年前，人们开始用电脑在互联网上查询信息、社交、购物；几年前，大多数人变成了低头一族，移动互联网让人们依靠一部智能手机，就能够在一个陌生的城市找到自己想去的任何地方。那么，在这样一个移动互联网的时代，响应式设计和 HTML5+CSS3 技术是我们必须要掌握的。本单元将带大家走进响应式设计和 HTML5+CSS3 的世界。

【教学导航】

学习目标	1. 了解响应式 Web 设计的概念 2. 了解 HTML5 和 CSS3 的特性和优势 3. 复习 HTML 和 CSS 的基础知识 4. 掌握 HTML5 的语义化结构标签 5. 对 HTML5 和 CSS3 的项目编码过程有初步的了解
教学方式	本单元以理论讲解、效果演示和代码解读为主。不要求学生对项目代码逐行理解，只需体验编码过程即可。需要注意的是： 项目 1-1，HTML 与 CSS 文件应该同步对应编写，每完成一个区块后，演示效果 项目 1-2，应先编写 HTML 结构，再编写 CSS 样式
重点知识	1. 响应式 Web 设计理念 2. 浏览器前缀，相关属性可参见附录 3 3. HTML5 的语义化结构标签
关键词	响应式、HTML5、语义化、浏览器前缀

响应式 Web 设计简介

越来越多的人使用小屏幕设备上网，针对不同屏幕的设备开发不同的页面成本非常大，这时，响应式 Web 设计应运而生。响应式 Web 设计（Responsive Web Design）是由 Ethan Marcotte 在 2010 年提出的，他将媒体查询、栅格布局和弹性图片合并称为响应式 Web 设计。

1. 设计理念

从设计理念看，响应式 Web 设计是一种针对任意设备对网页内容进行完美布局的显示方式，与原始设计方式相比有以下两点突破。

（1）一个网页设计，多个设备使用

随着移动产品的日益丰富，出现了各种屏幕尺寸的手机、Pad 等移动设备，而针对每一种尺寸的设备都独立开发一个网站，成本会非常高，如果要找一个成本、设计、性能的平衡点，响应式设计是最好的选择。它可以做到一处设计，响应多种屏幕。

（2）移动优先

以前的网站开发大多数是先开发 PC 端，再根据 PC 端的网页及功能设计开发移动端。然而，随着互联网行业的发展，使用移动端上网的用户群已经赶超 PC 端。由于移动端设备的屏幕小、计算资源低，如果我们先开发移动端，再开发 PC 端，可以迫使开发人员在更小、计算资源更低的设备中设计产品功能。这样做，一是可以使产品功能更加核心和简洁，二是有助于设计出性能更高的程序。

2. 用户体验

用户体验对于网站的运营是至关重要的，网站如果没有良好的用户体验，那么就算里面的内容再精彩，用户也无心浏览下去。通常，网站会在移动浏览器上缩放，这样虽然可以完整地给我们呈现想要浏览的内容，但鉴于移动设备屏幕大小的限制，过多的内容会使页面看起来杂乱不堪，用户也很难找到自己关注的内容。而响应式 Web 设计并不是将整个网页缩放给用户，而是经过

精心筛选，有选择性地显示页面的内容。

例如，一个用户个人信息界面在 PC 端大屏幕的页面效果如图 1-1 所示。

在图 1-1 中，该界面内容分三栏横向排列显示，如果在移动端的小屏幕上，按比例缩小，网页上的文字会看不清，使用响应式 Web 开发可以让该界面呈现纵向排列方式，如图 1-2 所示。

图1-1　个人信息界面PC端大屏幕效果

图1-2　个人信息界面移动端页面效果

3. 技术层面

在技术层面，响应式 Web 设计是和 HTML5+CSS3 互相配合与支持的，实现响应式设计包括以下技术点。

- HTML5+CSS3 的基本网页设计。
- HTML5 中的 viewport：提供可以配置视口的属性。
- CSS3 媒体查询（Media Queries）：识别媒体类型、特征（屏幕宽度、像素比等）。
- 流式布局（Fluid Layout）：可以根据浏览器的宽度和屏幕的大小自动调整效果。
- 响应式栅格系统（Responsesive Fluid Grid）：依赖于媒体查询，根据不同的屏幕大小调整布局。
- 流式图片（Fluid Images）：随流式布局进行相应缩放。

实现响应式 Web 设计，可以说就是根据显示屏幕大小的变化控制页面的文档流，那么学习响应式之前，必须有良好的 HTML5+CSS3 的页面开发基础，本教材在讲解如何实现页面的响应式之前，将为读者详细地讲解 HTML5+CSS3 的网页开发基础。

【项目 1-1】 使用 HTML5+CSS3 绘制 HTML5 的 logo

【项目描述】

HTML5 的官方 logo 如图 1-3 所示。该 logo 呈现出的感觉是强大、安全，具有前瞻性，耀眼明亮、用色大胆，而这些都代表着 HTML5 的形象。本项目将使用 HTML5+CSS3 绘制出 HTML5 的 logo，与此同时，通过完成本项目，读者会了解 HTML5 与 CSS3 给出的重大意义。

【前导知识】

HTML5 的新特性

HTML5 不仅仅是 HTML 规范的当前最新版本，也代表了一系列 Web 相关技术的总称，其中最重要的 3 项技术就是 HTML5 核心规范、CSS3（Cascading Style Sheet，层叠样式表的最新版本）和 JavaScript（一种脚本语言，用于增强网页的动态功能），这 3 项技术在后面的学习中会详细讲解。HTML 的历史可以追溯到很久以前，我们这里就不做讨论了。本书的关注点在于 HTML5 带给我们的全新感受。

1. 进化而非颠覆

试想，如果 HTML5 否定了之前的 HTML 文档，各种大大小小的网站都需要重新编写，那么 HTML5 带来的就不是惊喜而是惊吓！实际上，HTML5 的一个核心理念就是保持一切新特性与原有功能保持平滑过渡。在开发 HTML5 时，开发者还着重研究了以往 HTML 网页设计的一些通用行为，把代码重复率很高的功能提取为 HTML5 新标签，如<header>、<nav>等。

HTML5 进化的重大意义还在于，它从技术层面带来了 8 个类别的革新。用浏览器打开网址：http://www.w3.org/html/logo/，在该网站中将看到 HTML5 的 8 大革新 logo，如图 1-4 所示。

图1-3 HTML5官方logo

图1-4 HTML5的8大革新logo

图 1-4 中从左至右、从上到下，各 logo 的含义分别如下。

- 语义网（Semantics）：提供了一组丰富的语义化标签。
- 离线&存储（Offline & Storage）：HTML5 App Cache、Local Storage、Indexed DB 和 File API 使 Web 应用程序更加迅速，并提供了离线使用的能力。
- 设备访问（Device Access）：增强了设备感知能力，使得 Web 应用在电脑、Pad、手机上均能使用。
- 通信（Connectivity）：增强了通信能力，意味着增强了聊天程序的实时性和网络游戏的顺畅性。

- 多媒体（Multimedia）：音频视频能力的增强是 HTML5 的最大突破。
- 图形和特效（3D，Graphics & Effects）：Canvas、SVG 和 WebGL 等功能使得图形渲染更高效、页面效果更加炫酷。
- 性能和集成（Performance & Integration）：Web Worker 使浏览器可以多线程处理后台任务而不阻塞用户界面渲染。同时，性能检测工具方便评估程序性能。
- 呈现（CSS3）：CSS3 可以很高效地实现页面特效，并不会影响页面的语义和性能。

对于这 8 大革新，我们会在后续的学习中有更加深刻的体会。

2. 化繁为简

HTML5 以“简单至上，尽可能简化”为原则做了以下改进。

- 简化了 DOCTYPE 和字符集声明。
- 强化了 HTML5 API，使页面设计更加简单。
- 以浏览器的原生能力代替复杂的 JavaScript 代码。
- 精确定义的错误恢复机制，如果页面中有错误，也不会影响整个页面的显示。

3. 良好的用户体验

HTML5 规范以“用户至上”为宗旨。也就是说，在遇到冲突时，规范的优先级为：用户>页面作者>实现者（浏览器）>规范开发者（W3C/WHATWG）>纯理论。除此之外，HTML5 还引入了一种新的安全模型来保证 HTML5 足够安全。

对于页面设计而言，浏览器的支持情况至关重要，值得庆幸的是，各大浏览器对 HTML5 的支持正在不断完善，越来越多的开发者尝试在项目中使用 HTML5，特别是在移动互联网领域。目前，Chrome 对 HTML5 的支持最好，Firefox、Opera、Safari、IE10 对 HTML5 也有很好的支持。

本书推荐使用 Chrome 浏览器，因为它不仅简洁，而且附带便捷的开发工具。编写此书时 Chrome 浏览器的最高版本为 49.0.2623.87，为了使本教材中的项目呈现出最佳效果，请大家尽量使用最新版本。

HTML5 的基本语法

前面讲了很多关于 HTML5 的强大优势，接下来，我们就来使用 HTML5 进行网页设计。HTML 文档是由多种标签组成的，一个 HTML5 的标准模板如下所示。

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
</head>
<body>
    <!--这是注释 -->
</body>
</html>
```

下面对该模板的结构及组成进行详细的讲解。

（1）<!DOCTYPE>标签

<!DOCTYPE>标签位于文档的最前面，用于向浏览器说明当前文档使用的 HTML 版本，不可省略。HTML5 中使用 <!DOCTYPE html>声明，该声明方式适用于所有版本的 HTML，有兴

趣的读者可以查阅一下 HTML 其他版本的声明，HTML5 规范化繁为简的特性不言而喻。

（2）<html>标签

<html>标签标志着 HTML 文档的开始，</html>标签标志着 HTML 文档的结束，在它们之间的是文档的头部和主体内容。lang 属性规定元素内容的语言。

（3）<head>标签

<head>标签用于定义 HTML 文档的头部信息，主要用来封装其他位于文档头部的标签，如<title>、<meta>、<link>及<style>等，用来描述文档的标题、作者以及和其他文档的关系等。一个 HTML 文档只能含有一对<head>标签，绝大多数文档头部包含的数据不会作为内容显示在页面中。

（4）<body>标签

<body>标签用于定义 HTML 文档所要呈现的内容。浏览器中显示的所有文本、图像、音频和视频等信息都必须位于<body>标签内。一个 HTML 文档只能含有一对<body>标签，且<body>标签必须在<html>标签内，位于<head>头部标签之后，与<head>标签是并列关系。

（5）<!-- -->注释

<!-- -->中的内容用于对代码进行解释，不会呈现在页面上。

CSS3 的新特性

CSS 即层叠样式表（Cascading Style Sheet），主要用于设置 HTML 页面中的文本内容（字体、大小、对齐方式等）、图片的外形（宽高、边框样式、边距等）以及版面的布局等外观显示样式。

CSS3 是 CSS 的当前最新版本，该版本提供了更加丰富且实用的规范，如列表模块、超链接、语言模块、背景和边框、颜色、文字特效、多栏布局、动画等，这些规范的使用会在后面的单元中依次讲解。在 Web 开发中采用 CSS3 技术将会美化页面，创造动画效果，显著地提高用户体验，同时也能极大地提高程序的性能。

如何在 HTML 中引入样式表

1. 行内式

行内式是通过标签的 style 属性来设置元素的样式，其基本语法格式如下。

```
<标签名 style="属性 1:属性值 1; 属性 2:属性值 2; 属性 3:属性值 3;"> 内容 </标签名>
```

任何 HTML 标签都拥有 style 属性，用来设置行内式。

2. 内嵌式

内嵌式是将 CSS 代码集中写在 HTML 文档的<head>头部标签中，并且用<style>标签定义，其基本语法格式如下。

```
<style>
    选择器 {属性 1:属性值 1; 属性 2:属性值 2; 属性 3:属性值 3;}
</style>
```

<style>标签一般位于<head>标签中<title>标签之后，由于浏览器是从上到下解析代码的，把 CSS 代码放在头部便于提前被下载和解析。

3. 链入式

链入式是将所有的样式放在一个或多个以.css 为扩展名的外部样式表文件中，通过<link/>标签将外部样式表文件链接到 HTML 文档中，其基本语法格式如下。

```
<link href="CSS 文件的路径" type="text/css" rel="stylesheet" />
```

该语法中，<link />标签需要放在<head>头部标签中，并且指定<link />标签的 3 个属性，具体如下。

- href：定义所链接外部样式表文件的 URL，可以是相对路径，也可以是绝对路径。
- type：定义所链接的文档类型，“text/css”表示链接的外部文件为 CSS 样式表。
- rel：定义当前文档与被链接文档之间的关系，在这里需要指定为“stylesheet”，表示被链接的文档是一个样式表文件。

CSS3 与浏览器

为了更好地兼容不同内核的浏览器，CSS3 中部分属性需要添加浏览器的私有前缀，具体如下。

- 以-webkit-开头的样式，只有以 Webkit 为内核的浏览器可以解析，如 Chrome、Safari。
- 以-moz-开头的样式，只有以 Gecko 为内核的浏览器可以解析，如 Firefox。
- 以-ms-开头的样式，只有以 Trident 为内核的浏览器可以解析，如 IE。
- 以-o-开头的样式，只有以 Presto 为内核的浏览器可以解析，如 Opera。

CSS3 中需要添加浏览器私有前缀的属性，可参见附录 3。

【项目分析】

本项目是由 div 块级元素拼接而成，用 CSS 样式控制每一块的样式和位置。完成本项目需要 8 个步骤，如图 1-5 所示。

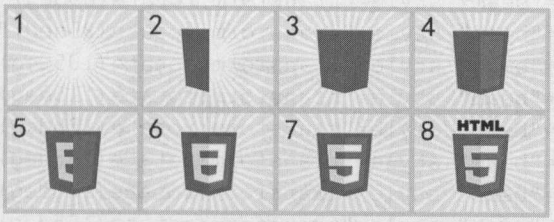


图1-5 步骤图

从图 1-5 可以看出页面的实现步骤，具体如下。

（1）定位出整个页面的背景区域“bg”，并实现背景光束。

（2）定义 logo 样式，并画出盾牌的左半边。

（3）画出盾牌的右半边。

（4）画出浅橘色区域。

（5）画出“5”的左半边。

（6）画出“5”的右半边。

（7）用色块遮盖多余的部分。

（8）在盾牌上方添加“HTML”图片。

【代码实现】

这是一个简单而有趣的项目，接下来用代码来实现该页面，该项目的 HTML 代码如 code\01\0101\0101.html 所示。

0101.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<!--网页标题为 HTML5 logo-->
<title>HTML5 logo</title>
<!--链入式样式表，浏览器运行时会加载解析 css/0101/目录下的 logo.css 文件-->
<link href="css/logo.css" type="text/css" rel="stylesheet">
</head>
<body>
  <!--这是 HTML5 logo 最外层的盒子 div，类名为 bg-->
  <div class="bg" >
    <!--每一束白色光束的 div 盒子，行内样式 style="transform:rotate(5deg)"用于设置 div 变形的样式，即旋转 5 度，这是 CSS3 新增的属性-->
    <div class="beam" style="transform:rotate(5deg)"></div>
    <div class="beam" style="transform:rotate(15deg)"></div>
    <div class="beam" style="transform:rotate(25deg)"></div>
    <div class="beam" style="transform:rotate(35deg)"></div>
    <div class="beam" style="transform:rotate(45deg)"></div>
    <div class="beam" style="transform:rotate(55deg)"></div>
    <div class="beam" style="transform:rotate(65deg)"></div>
    <div class="beam" style="transform:rotate(75deg)"></div>
    <div class="beam" style="transform:rotate(85deg)"></div>
    <div class="beam" style="transform:rotate(95deg)"></div>
    <div class="beam" style="transform:rotate(105deg)"></div>
    <div class="beam" style="transform:rotate(115deg)"></div>
    <div class="beam" style="transform:rotate(125deg)"></div>
    <div class="beam" style="transform:rotate(135deg)"></div>
    <div class="beam" style="transform:rotate(145deg)"></div>
    <div class="beam" style="transform:rotate(155deg)"></div>
    <div class="white beam" style="transform:rotate(165deg)"></div>
    <div class="white beam" style="transform:rotate(175deg)"></div>
    <!--中心 logo 的 div 盒子类名为 logo，设置行内样式为绝对定位，距离上边 120px,距离左边 229px-->
    <div class="logo" style="top:120px;left:229px;">
      <!-- 盾牌（左） -->
      <div class="d_shield1" ></div>
      <div class="d_shield2" ></div>
      <div class="d_shield3" ></div>
      <!-- 盾牌（右） -->
      <div class="d_shield4" ></div>
      <div class="d_shield5" ></div>
      <div class="d_shield6" ></div>
      <!-- 浅橘色部分 -->
      <!--定义行内样式给整个浅橘色部分的 div 盒子进行某种变形（CSS3 新增属性），缩放为原来的 0.82 倍，并给这个 div 绝对定位-->
      <div style="transform:scale(0.82);left:31px;top:25px">
        <div class="l_shield1" ></div>
```

```
46          <div class="l_shield2" ></div>
47          <div class="l_shield3" ></div>
48        </div>
49        <!--以下 4 个 div 组成的 logo 里面数字 5 的左半边灰色部分-->
50        <div class="gray1" ></div>
51        <div class="gray2" ></div>
52        <div class="gray3" ></div>
53        <div class="gray4" ></div>
54        <!--以下 4 个 div 组成的 logo 里面数字 5 的右半边白色部分-->
55        <div class="white1" ></div>
56        <div class="white2" ></div>
57        <div class="white3" ></div>
58        <div class="white4" ></div>
59        <!-- 最后的修补 -->
60        <div class="d_shield7" ></div>
61        <div class="l_shield4" ></div>
62        <img src="images/HTML.png">
63      </div>
64    </div>
65 </body>
66 </html>
```

样式代码如 code\01\0101\css\logo.css 所示。

logo.css

```
1  /*第 1 单元 项目 1-1  使用 HTML5+CSS3 绘制 HTML5 的 logo*/
2  body{
3     margin:0;   /*设置 body 的外边距为 0*/
4     padding:0;  /*设置 body 的内边距为 0*/
5  }
6  div{
7     position:absolute;/*给所有的 div 设置绝对定位*/
8  }
9  /*给类名为 bg 的元素设置宽、高、背景颜色*/
10 .bg{
11    width:800px;
12    height:600px;
13    background:#f2f2f2;
14    overflow:hidden;/*隐藏内容溢出*/
15 }
16 /*给类名为 beam 的元素设置宽、高、背景颜色及绝对定位*/
17 .beam{
18    width:1600px;/*宽度为 1  600px*/
19    height:20px;/*高度为 20px*/
20    background:#fff;/*背景为白色*/
21    top:290px;/*绝对定位，距父元素上边线 290px*/
22    left:-400px;/*绝对定位，距父元素左边线-400px*/
23 }
24 .d_shield1,.d_shield2,.d_shield3,.d_shield4,.d_shield5,.d_shield6,
25 .d_shield7{
26    background:#e15016;
```

```
}
/*给类名为 d_shield1 的元素设置宽、高及绝对定位*/
.d_shield1{
    left:32px;/*绝对定位，距父元素左边线 32px*/
    width:140px;
    height:346px;
}
.d_shield2{
    transform:skewx(5deg);/*变形（CSS3 新增属性）：水平方向斜切 5 度*/
    left:16px;/*绝对定位，距父元素左边线 16px*/
    width:100px;
    height:346px;
}
.d_shield3{
    transform:skewy(15deg);/*变形：垂直方向斜切 15°*/
    top:265px;/*绝对定位，距父元素上边线 265px*/
    left:32px;/*绝对定位，距父元素左边线 32px*/
    width:140px;
    height:100px;
}
.d_shield4{
    left:172px;
    width:140px;
    height:346px;
}
.d_shield5{
    transform:skewx(-5deg);
    left:227px;
    width:100px;
    height:346px;
}
.d_shield6{
    transform:skewy(-15deg);
    top:265px;
    left:172px;
    width:140px;
    height:100px;
}
.d_shield7{
    height:20px;
    top:199px;
    width:80px;
    left:60px;
}
.l_shield1,.l_shield2,.l_shield3,.l_shield4{
    background:#ee6812;
}
.l_shield1{
    left:172px;
    width:140px;
```

```
77      height:346px;
78  }
79  .l_shield2{
80      transform:skewx(-5deg);
81      left:227px;
82      width:100px;
83      height:363px;
84  }
85  .l_shield3{
86      transform:skewy(-15deg);
87      top:282px;
88      left:172px;
89      width:138px;
90      height:100px;
91  }
92  .l_shield4{
93      height:43px;
94      top:113px;
95      width:100px;
96      left:180px;
97  }
98  .gray1,.gray2,.gray3,.gray4{
99      background:#ebebeb;
100 }
101 /*给类名为 gray1 的元素设置宽、高及绝对定位*/
102 .gray1{
103     height:43px;
104     width:102px;
105     left:70px;
106     top:70px;
107 }
108 /*给类名为 gray2 的元素设置宽、高、斜切变形及绝对定位*/
109 .gray2{
110     width:46px;
111     height:216px;
112     transform:skewx(5deg);
113     top:70px;
114     left:75px;
115 }
116 .gray3{
117     width:95px;
118     height:43px;
119     left:77px;
120     top:156px;
121 }
122 .gray4{
123     width:87px;
124     height:47px;
125     left:85px;
126     top:251px;
```

```
127    transform:skewy(15deg);
128}
129/*给类名为 white1、white2、white3、white4 的 4 个元素都设置背景颜色为白色*/
130.white1,.white2,.white3,.white4{
131    background:#fff;
132}
133/*给类名为 white1 的元素设置宽、高及绝对定位*/
134.white1{
135    width:102px;
136    height:43px;
137    left:172px;
138    top:70px;
139}
140/*给类名为 white2 的元素设置宽、高、斜切变形及绝对定位*/
141.white2{
142    width:46px;
143    height:216px;
144    transform:skewx(-5deg);
145    top:70px;
146    left:223px;
147}
148.white3{
149    height:43px;
150    width:95px;
151    left:172px;
152    top:156px;
153}
154.white4{
155    height:47px;
156    width:87px;
157    left:172px;
158    top:251px;
159    transform:skewy(-15deg);
160}
161/*给 img 标签设置宽、高及固定定位*/
162img {
163    position: fixed;/ *固定定位*/
164    top: 8px;
165    left: 225px;
166    width: 350px;
167    height: 110px;
168}
```

【项目总结】

本项目的练习重点：

通过本项目的练习，读者应该了解 HTML5 和 CSS3 在前一版本基础上的优势和进化部分，并且在练习编码的过程中，体会 HMTL5 和 CSS3 的新特性。

本项目的练习方法：

建议读者在编码时按照【项目分析】中的 8 个步骤，采用 HTML 和 CSS 同步编写的方法。例如，首先编写 HTML 文件中的整体结构，class 为 bg、beam、white beam 的 标签；然后在 CSS 文件中添加整体样式（body、div）和背景样式（bg、beam、white beam）；编写完成后保存文件，并用浏览器打开页面，即可呈现出前面步骤中所说的白色光束。后面步骤的编写方法以此类推。

本项目的注意事项：

如果读者编辑的页面在浏览器中显示样式错乱，解决办法有两个。

一是将浏览器升级为最高版本。

二是将代码中所有的 transform 属性前加上对应浏览器的私有前缀，如谷歌浏览器加-webkit-前缀。

【项目 1-2】 构建移动版旅游网站页面

【项目描述】

要构建网页，首要考虑的就是网页的整体结构。本项目将使用 HTML5 语义化结构标签来构建一个简单的页面。与此同时，也通过本项目来回顾一下 CSS 的基础知识。页面效果如图 1-6 所示。

图1-6　移动版旅游网站页面

【前导知识】

HTML5 语义化结构标签

HTML5 定义了一种新的语义化标签来描述元素的内容，让很多更语义化的结构化代码标签代替大量无意义的 标签。下面列举一些 HTML5 中常用的语义化标签，如表 1-1 所示。

表 1-1　语义化标签

标签名	描　述
<header>	表示页面中一个内容区块或整个页面的标题
<section>	页面中的一个内容区块，如章节、页眉、页脚或页面的其他部分，可以和 h1、h2...元素结合起来使用，表示文档结构
<article>	表示页面中一块与上下文不相关的独立内容，如一篇文章
<aside>	表示<article>标签内容之外的、与<article>标签内容相关的辅助信息
<hgroup>	表示对整个页面或页面中的一个内容区块的标题进行组合
<figure>	表示一段独立的流内容，一般表示文档主体流内容中的一个独立单元
<figcaption>	定义<figure>标签的标题
<nav>	表示页面中导航链接的部分
<footer>	表示整个页面或页面中一个内容区块的脚注。一般来说，它会包含创作者的姓名、创作日期及联系方式

在本旅游网站项目中，传统方式布局与 HTML5 语义化标签布局的对比如图 1-7 和图 1-8 所示。

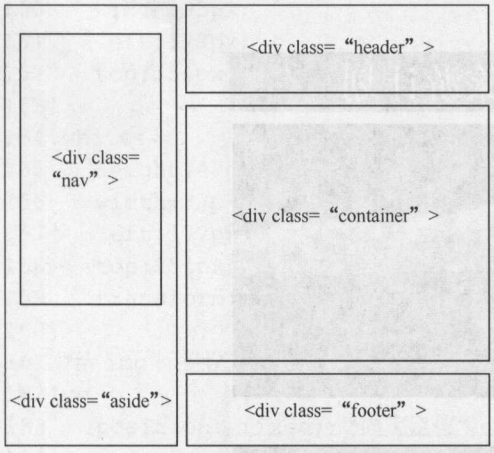


图1-7　采用div+CSS布局页面

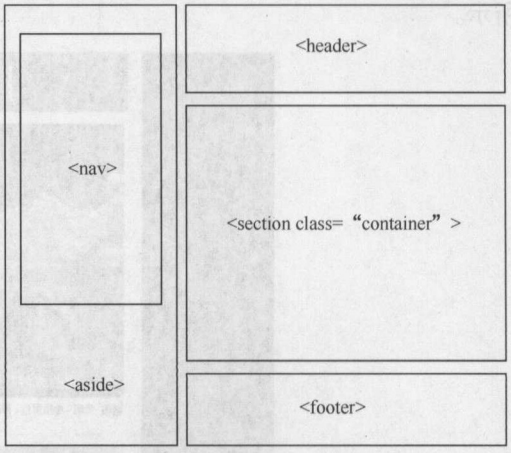


图1-8　采用HTML5新标签布局页面

在图 1-8 中，<section>标签经常用来表示一段专题性的内容，一般会带有标题，它的典型应用场景有文章的章节、标签对话框中的标签页等。虽然语义化不会让用户马上感受到它的好处，但语义化标签不仅提升了网页的质量和语义，而且对搜索引擎能起到良好的优化效果。

除上述标签外，HTML5 新增了很多标签，这些标签会在后面的章节中分类讲解。

注意

HTML 中的标签与元素

在 HTML 中，标签以尖括号包围着；元素由开始标签、结束标签和中间的内容 3 部分组成，如<title>Document</title>，title 为元素名。

CSS 选择器

在 CSS 中，要想将 CSS 样式应用于特定的 HTML 标签，首先需要找到该目标标签。CSS 选择器的作用就是从 HTML 页面中找出特定的某类元素。常用的几类 CSS 选择器如表 1-2 所示。

表 1-2　CSS 选择器

选择器	代码	示例代码	说明
通用选择器	*	*{}	选择所有元素
标签选择器	元素名称	a{}、body{}、p{}	根据标签选择元素
类选择器	.<类名>	. beam {}	根据 class 的值选择元素
id 选择器	#<id 值>	#logo{}	根据 id 的值选择元素
属性选择器	[<条件>]	[href]{}、[attr="val"]{}	根据属性选择元素
并集选择器	<选择器>,<选择器>	em,strong{}	同时匹配多个选择器，取多个选择器的并集
后代选择器	<选择器> <选择器>	.asideNav li {}	先匹配第二个选择器的元素，并且属于第一个选择器内
子代选择器	<选择器> > <选择器>	ul>li{}	匹配第二个选择器，且为第一个选择器内元素的后代
兄弟选择器	<选择器>+<选择器>	p+a{}	匹配紧跟第一个选择器，并且匹配第二个选择器内的元素，如紧跟 p 元素后的 a 元素
伪选择器	: :<伪元素>或 : <伪类>	p::first-line{}、a:hover{}	伪选择器不是直接对应 HTML 中定义的元素，而是向选择器增加特殊的效果

其中，伪选择器比较特殊，分为伪元素和伪类两种。常用的伪元素选择器如表 1-3 所示。

表 1-3　伪元素选择器

元素名	描　　述
::first-line	匹配文本块的首行，如 p::first-line 表示选中 p 元素的首行
::first-letter	匹配文本内容的首字母
::before	在选中元素的内容之前插入内容
::after	在选中元素的内容之后插入内容

常用的伪类选择器如表 1-4 所示。

表 1-4　伪类选择器

元素名	描　　述
:root	选择文档中的根元素，通常返回 html
:first-child	父元素的第一个子元素
:last-child	父元素的最后一个子元素
:only-child	父元素有且只有一个子元素
:only-of-type	父元素有且只有一个指定类型的元素
:nth-child(n)	匹配父元素的第 n 个子元素
:nth-last-child(n)	匹配父元素的倒数第 n 个子元素
:nth-of-type(n)	匹配父元素定义类型的第 n 个子元素
:nth-last-of-type(n)	匹配父元素定义类型的倒数第 n 个子元素

续表

元素名	描　述
:link	匹配链接元素
:visited	匹配用户已访问的链接元素
:hover	匹配处于鼠标悬停状态下的元素
:active	匹配处于被激活状态下的元素，包括即将单击（按压）
:focus	匹配处于获得焦点状态下的元素
:enabled (:disabled)	匹配启用（禁用）状态的元素
:checked	匹配被选中的单选按钮和复选框的 input 元素
:default	匹配默认元素
:valid (:invalid)	根据输入数据验证，匹配有效（无效）的 input 元素
:in-range (out-of-range)	匹配在指定范围之内（之外）受限的 input 元素

列举了这么多选择器，下面演示几个选择器的用法，如 demo1-1 所示。

demo1-1.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>选择器的使用</title>
</head>
<style type="text/css">
    /*类型选择器 nav 表示导航*/
    nav {
        width:300px;
        height:400px;
    }
    /*类型选择器 a 表示链接*/
    a {
        text-decoration:none;
        display:block;
        color: #fff;
        height:35px;
        padding:0 38px;
    }
    li {
         background-color: rgba(0,0,0,0.7);/*背景颜色透明度为 0.7*/
         height:35px;
         line-height: 35px;
         overflow: hidden;
    }
    /*偶数行背景颜色透明度为 0.9*/
    li:nth-of-type(2n) {
         background-color: rgba(0,0,0,0.9);
    }
    /*鼠标悬停时背景颜色为#0099E5*/
```

```
    li:hover {
        background: #0099E5;
    }
</style>
<body>
    <nav>
     <ul>
      <li><a href="#" ><span>JavaEE 培训</span></a></li>
      <li><a href="#" ><span>Android 培训</span></a></li>
      <li><a href="#" ><span>PHP 培训</span></a></li>
      <li><a href="#" ><span>UI 设计培训</span></a></li>
      <li><a href="#" ><span>iOS 培训</span></a></li>
      <li><a href="#" ><span>前端与移动开发培训</span></a></li>
      <li><a href="#" ><span>C/C++培训</span></a></li>
      <li><a href="#" ><span>网络营销培训</span></a></li>
      <li><a href="#" ><span>游戏开发培训</span></a></li>
      <li><a href="#" ><span>云计算之大数据培训</span></a></li>
     </ul>
    </nav>
</body>
</html>
```

用浏览器打开 demo1-1，页面效果如图 1-9 所示。

在 demo1-1 中，使用了类型选择器和伪类选择器实现了一个导航侧边栏的效果，其中，用 nth-of-type(2n)选择器实现了奇偶行背景颜色差异，用：hover 选择器实现了鼠标悬停背景颜色变化的功能。其他选择器的使用方法与此类似，读者可以自行尝试。

盒子模型

CSS 中的一个基本概念就是盒子模型。所谓盒子模型就是将 HTML 页面中的元素视为一个矩形区域，即元素的盒子。盒子由 margin（外边距）、border（边框）、padding（内边距）和 content（内容）4 部分组成，盒子的各属性如图 1-10 所示。

图1-9　demo1-1页面效果

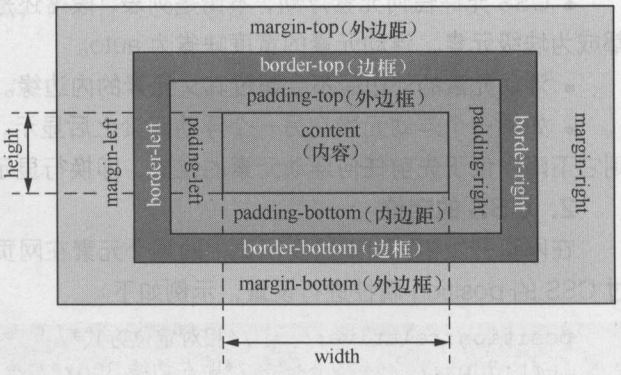


图1-10　盒子的各属性

网页是由多个元素构成的盒子排列而成的。而多个盒子之间会出现外边距合并的现象，下面总结了在网页制作时关于多个盒子之间需要注意的问题。

• 相邻块级元素的垂直外边距合并：例如，上下相邻的块元素，如果上面的元素有下外边距，下面的元素有上外边距，则垂直边距为两者中的较大者，如图 1-11 所示。

• 嵌套块级元素的垂直外边距合并：例如，父元素没有上内边距和边框，则父元素与子元素的上外边距合并为较大者，如图 1-12 所示。

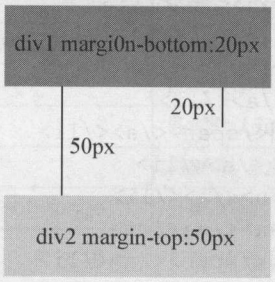


图1-11　相邻块级元素

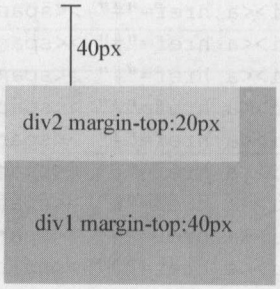


图1-12　嵌套块级元素

CSS 的浮动与定位

在一个网页中，默认情况下块级元素独占一行，自上而下排列，行内元素自左向右排列。但是，在实际的网页布局中往往需要改变这种单调的排列方式，而使网页内容变得丰富多彩。CSS 的浮动和定位完美地解决了这个问题。

1. CSS 的浮动

CSS 的浮动可以通过 float 属性进行设置，float 的常用属性值如表 1-5 所示。

表 1-5　float 的常用属性值

属性值	描　述
left	元素向左浮动
right	元素向右浮动
none	元素不浮动（默认值）

CSS 元素的浮动需要注意以下几点。

• CSS 允许任何元素浮动，不论是列表、段落还是图像。无论元素先前是什么状态，浮动后都成为块级元素，浮动元素的宽度缺省为 auto。

• 浮动元素的外边缘不会超过其父元素的内边缘。

• 如果一个浮动元素在另一个浮动元素之后显示，而且会超出容纳块（没有足够的空间），则它下降到低于先前任何浮动元素的位置，即换行显示。

2. CSS 的定位

在网页开发中，如果需要网页中的某个元素在网页的特定位置出现，如弹出菜单，则可以通过 CSS 的 position 属性进行设置，示例如下。

```
position:relative;      /*相对定位方式*/
left:30px;              /*距左边线 30px*/
top:10px;               /*距顶部边线 10px*/
```

用于设置定位方式的常用属性值如表 1-6 所示。

表 1-6　position 属性定位方式的常用值

值	描　述
static	静态定位（默认定位方式）
relative	相对定位，相对于其原文档流的位置进行定位
absolute	绝对定位，相对于 static 定位以外的第一个上级元素进行定位
fixed	固定定位，相对于浏览器窗口进行定位

用于设置元素具体位置的常用属性值如表 1-7 所示。

表 1-7　position 属性边偏移设置方式

边偏移属性	描　述
top	顶端偏移量，定义元素相对于其参照元素上边线的距离
bottom	底部偏移量，定义元素相对于其参照元素下边线的距离
left	左侧偏移量，定义元素相对于其参照元素左边线的距离
right	右侧偏移量，定义元素相对于其参照元素右边线的距离

3. 浮动和定位的使用区别

其实，浮动的本意是用来解决图片和文字的排版问题，但是由于它十分好用，所以被大部分开发者应用到了网页布局中，并成为了公认布局的一种方式。

接下来，我们通过一个 demo 演示一下浮动和定位在网页布局中的使用，如 demo1-2 所示。

demo1-2.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>浮动和定位</title>
</head>
<style type="text/css">
    /*给类名为 header、aside、main、footer 的元素设置背景颜色和边框*/
    .header,.aside,.main,.footer{
        background-color: pink; /*背景色为粉色*/
        border: 1px solid yellow; /*边框为 1px 的黄色实线*/
    }
    .header{
        height: 100px;
    }
    .aside,.main{
        height: 200px;
    }
    .aside{
        width: 200px;
        float:left;  /*类名为 aside 的元素左浮动*/
    }
    .main{
```

```
        margin-left: 202px; /*元素左边距为 202px*/
    }
    .footer{
        height: 50px;
    }
    /*给类名为 float-div 的元素设置背景颜色、边框、宽高及绝对定位*/
    .float-div{
        background-color: paleturquoise; /*背景色为苍白的宝石绿*/
        border: 1px solid yellow; /*边框为 1px 的黄色实线*/
        width: 100px;
        height:100px;
        position: absolute; /*绝对定位*/
        top: 160px; /*距父元素顶部边线 160px*/
        left: 500px; /*距父元素左边线 500px*/
    }
</style>
<body>
    <header class="header">header</header>
    <aside class="aside">aside</aside>
    <section class="main">section</section>
    <footer class="footer">footer</footer>
    <div class="float-div">floatdiv</div>
</body>
</html>
```

用浏览器打开 demo1-2，页面效果如图 1-13 所示。

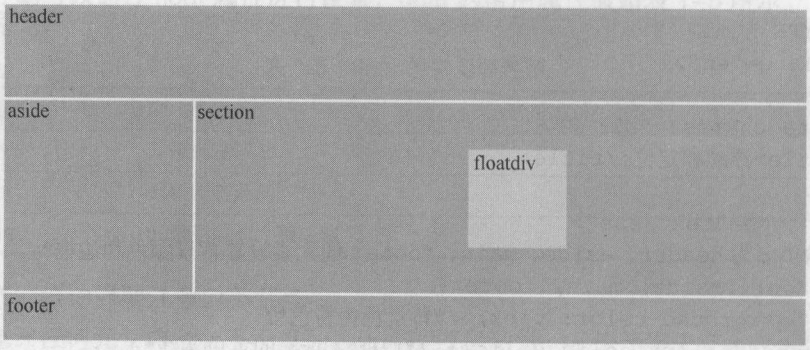


图1-13　demo1-2页面效果

在 demo1-2 中，使用浮动的知识对页面进行了布局，也就是图 1-13 中的 header、aside、section、footer 区域，然后使用绝对定位知识创建了一个浮动的 div 元素 floatdiv。需要注意的是，position: absolute 会导致元素脱离文档流，被定位的元素等于在文档中不占据任何位置，在另一个层呈现。float 也会导致元素脱离文档流，但还在文档或容器中占据位置，把文档流和其他 float 元素向左或向右挤，并可能导致换行。

4. z-index 层叠等级属性

当一个父元素中的多个子元素同时被定位时，定位元素之间有可能会发生重叠，如图 1-14 所示。

我们知道，显示器显示的图案是一个二维平面，使用 x 轴和 y 轴来表示位置属性。为了表示三维立体的概念，如图 1–14 中上下层的立体关系，引入了 z–index 属性来表示 z 轴的深度。z–index 值可以控制定位元素在垂直于显示屏方向（z 轴）上的堆叠顺序，值大的元素发生重叠时会在值小的元素上面，其取值可为正整数、负整数和 0，默认值为 0。

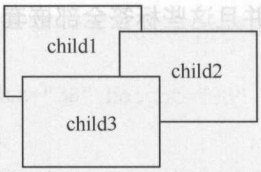


图1–14　定位元素发生重叠

注意

z–index 只能在 position 属性值为 relative、absolute 或 fixed 的元素上有效，z 轴可以理解为屏幕的深度，z–index 值越大的元素越靠近用户。

【项目分析】

有了前导知识作为铺垫，接着我们来分析一下本项目的页面结构。页面标注和页面结构如图 1–15 和图 1–16 所示。

图1–15　页面标注

图 1–16 所示的旅游网站页面，主要由侧边栏和主体部分两部分构成。

该页面的实现细节具体如下。

（1）侧边栏使用<aside>标签嵌套<nav>标签来实现，在<nav>标签中嵌套<ul>列表实现导航菜单。

（2）页面右侧部分从上至下依次使用<header>标签、<section>标签和<footer>标签来实现，

并且这些标签全部嵌套在一个<section>中。

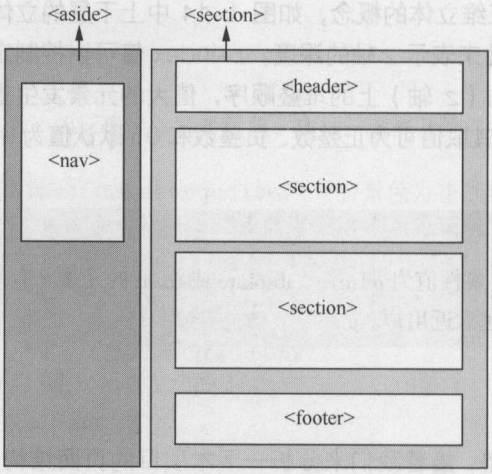


图1-16　页面结构

（3）<header>标签中文字使用<h1>标签，并设置样式。

（4）两个旅游信息的<section>中分别嵌套两个 标签，用来设置旅游图片和文字信息在页面中的位置。

（5）<footer>标签中使用<p>、和<a>链接来添加一些文字和链接信息。

【代码实现】

该项目的 HTML 页面代码如 code\01\0102\ 0102.html 所示。

0102.html

```
<!DOCTYPE html>
<html lang="zh-cn">
 <head>
  <meta charset="utf-8" />
  <title>国内旅游</title>
  <!--链入对网页某类型元素的通用设置的样式表-->
  <link href="css/common.css" rel="stylesheet" type="text/css"/>
  <!--链入对某个元素的特定设置的样式表-->
  <link href="css/main.css" rel="stylesheet" type="text/css" />
 </head>
 <body>
  <!--右半边-->
  <section class="qui-page">
   <!--header-->
   <header class="qui-header">
      <h1>国内旅游计划</h1>
   </header>
   <!--/header-->
   <!--begin-->
   <section class="container"> <!--表示页面中的一个内容区块-->
     <div  class="plcRouteList">
```

```
        <a href="#"> <img src="images/fengjing.jpg" width="100%" alt="" class="pic" />
        <div class="bottom ">
          <p class="face"><img src="images/touxiang.jpg" width="38" height="38"
          alt="" /></p>
          <h2 class="title">我的旅游行程</h2>
        </div>
        <p class="day">14 天</p> </a>
        <div class="infos">
         <div>
          <em>城市</em>  <!--<em>标签表示其中的文本为强调的内容-->
          <p> 昆明 - 香格里拉 - 西藏 </p>
         </div>
        </div>
      <!--</li>-->
     <!--</ul>-->
    </div>
  </section>
  <!--end-->
  <!--begin-->
  <section class="container"> <!--语义化元素表示页面中的一个内容区块-->
   <div class="plcRouteList">
     <a href="#"> <img src="images/华东.jpg" width="100%" alt="" class="pic" />
      <div class="bottom ">
       <p class="face"><img src="images/touxiang.jpg" width="38" height= "38"
       alt="" /></p>
       <h2 class="title">我的旅游行程</h2>
      </div>
      <p class="day">15 天</p> </a>
     <div class="infos">
      <div>
       <em>城市</em>  <!--<em>标签表示其中的文本为强调的内容-->
       <p> 北京 - 常州 - 苏州 </p>
      </div>
     </div>
   </div>
  </section>
  <!--footer-->
  <footer class="qui-footerBasic">  <!--表示页面或页面中一个内容区块的脚注-->
   <p class="switchStyle"><span >手机版</span><a  href="#"><span>电脑版
   </span></a><span><a href="#">APP</a></span></p>
  </footer>
  <!--footer end-->
 </section>
 <!--右半边 end-->
 <!--侧边栏导航-->
 <!--表示 article 元素内容之外的、与 article 元素内容相关的辅助信息-->
 <aside class="qui-asides">
  <section class="qui-aside">
   <nav class="qui-asideNav">  <!--表示页面中导航链接的部分-->
```

```
68      <ul>
69       <li><a href="#" ><span>首页</span></a></li>
70       <li><a href="#" ><span>目的地</span></a></li>
71       <li><a href="#" ><span>酒店</span></a></li>
72       <li><a href="#" ><span>机票</span></a></li>
73       <li><a href="#" ><span>时间</span></a></li>
74       <li><a href="#" ><span>点评</span></a></li>
75      </ul>
76     </nav>
77     </section>
78   </aside>
79   <!--侧边栏导航 end-->
80  </body>
81 </html>
```

在上述代码中，标签表示其中的文本为强调的内容，这段文字显示为斜体。当然，我们可以在样式表中清除其字体样式，重新设置其他字体样式。

该项目的 CSS 代码分为两部分，一部分是对网页某类型元素的通用设置，另一部分是对某个元素的特定设置，通用样式设置如 code\01\0102\css\common.css 所示。

common.css

```
1  /*第 1 单元 项目 1-2  旅游网站页面 common.css*/
2  html {
3      height: 100%;
4      overflow-x: hidden; /*表示水平方向隐藏溢出，没有滚动条*/
5      background: #f5f5f5;
6      color: #444;
7     /*设置字体样式。!important 设置该元素的样式具有最高权值*/
8      font: 14px/24px Helvetica !important;
9  }
10 body {
11     -webkit-box-sizing: border-box;
12     -moz-box-sizing: border-box;
13    /*为元素指定的任何内边距和边框都将在已设定的宽度和高度内进行绘制*/
14     box-sizing: border-box;
15     position: relative;
16     z-index: 0;
17     width: 100%;
18     max-width: 640px;
19     min-height: 100%;
20     margin: 0 auto; /*页面水平居中*/
21     overflow-x: hidden;
22    /*盒子阴影：水平偏移，垂直偏移模糊值，阴影颜色（其中 rgb 指颜色，a 指透明度*/
23     box-shadow: 0 0 10px rgba(0,0,0,0.3);
24 }
25 div,ul,li,p {
26     margin: 0;
27     padding: 0;
28     outline: none; /*当元素获得焦点的时候，焦点框为 0，不出现虚线框（或高亮框）*/
29 }
```

```
em,strong {
    font-style: normal; /*字体样式正常*/
    font-weight: normal; /*字体粗细正常*/
}
ul {
    list-style: none; /*清除默认样式*/
}
h1 {
    font-size: 55px; /*字体大小 55px*/
    margin-top: 30px; /*上边距 30px*/
    color: white; /*字体颜色为白色*/
    text-align: center;/ *文字水平居中*/
}
```

主要样式设置如 code\01\0102\css\main.css 所示。

main.css

```
/*第 1 单元 项目 1-2  旅游网站页面 main.css*/
.qui-page {
    width: 640px;
}
.qui-header {
    width: 100%;
    height: 80px;
    overflow: hidden;  /*隐藏溢出内容*/
    background-color: #2bab79;  /*背景颜色为#2bab79*/
}
.container {
    width: 100%;
    -webkit-box-sizing: border-box;
    -moz-box-sizing: border-box;
    box-sizing: border-box;
}
.plcRouteList {
    border-bottom: 1px solid #e6e8ea;  /*底部边框为 1px、颜色为#e6e8ea 的实线 */
    background-color: #fff;
    padding-left: 5px;  /*左内边距为 5px*/
}
.plcRouteList li {
    padding: 15px 5px 15px 0;  /*上、右、下、左内边距分别为 15px、5px、15px、0*/
    border-top: 1px solid #e6e8ea;  /*顶部边框为 1px、颜色为#e6e8ea 的实线 */
}
.plcRouteList a {
    display: block;  /*将 a 由行元素变为块元素*/
    position: relative;
}
.plcRouteList .pic {
    display: block;
    min-height: 150px;
}
.plcRouteList .bottom {
```

```
35      position: absolute;
36      left: 0;
37      right: 0;
38      bottom: 0;
39      height: 50px;
40      padding: 30px 0 0 60px;
41      color: #fff;
42      /*背景颜色：线性渐变，从上到下由黑色渐变至透明度为 60%的黑色*/
43      background-image: -webkit-linear-gradient(top, rgba(0,0,0,0), rgba
        (0,0,0,0.6));
44      background-image: linear-gradient(top, rgba(0,0,0,0), rgba(0,0,0,0.6));
45  }
46  .plcRouteList .bottom .face {
47      float: left;  /*向左浮动*/
48      margin-left: -50px;
49      width: 38px;
50      height: 38px;
51      border: 1px solid #fff;
52      border-radius: 50%;  /*为元素添加圆角边框，边框半径为宽度的 50%*/
53      overflow: hidden;
54  }
55  .plcRouteList .bottom .face img {
56      display: block;
57      border-radius: 50%;
58  }
59  .plcRouteList .bottom  .title {
60      width: 100%;
61      overflow: hidden;
62      text-overflow: ellipsis;/*表示当对象内文本溢出时显示省略标记（...）*/
63      white-space: nowrap;
64      font-size: 18px;
65      font-weight: bold;
66      line-height: 22px;  /*行高为 22px*/
67  }
68  .plcRouteList .day {
69      position: absolute;
70      top: 10px;
71      right: 10px;
72      width: 50px;
73      height: 50px;
74      /*背景颜色：rgb 对应的是颜色值，a 对应的是透明度*/
75      background-color: rgba(43,171,121,0.8);
76      border-radius: 50%;
77      text-align: center;
78      font-size: 18px;
79      line-height: 50px;
80      color: #fff;
81  }
82  .plcRouteList .infos {
83      margin-top: 7px;  /*顶部外边距为 7px*/
```

```
84 }
85 .plcRouteList .infos>div {
86     margin-bottom: 5px;  /*底部外边距为 5px*/
87     padding-left: 40px;  /*左边内边距为 40px*/
88 }
89 .plcRouteList .infos em {
90     float: left;
91     margin-left: -40px;
92     font-size: 15px;
93     font-weight: bold;
94 }
95 .plcRouteList .infos p {
96     max-height: 48px;
97     overflow: hidden;
98 }
99 .qui-footerBasic {
100    width: 100%;
101    margin: 20px 0;  /*上下外边距为 20px，左右外边距为 0*/
102    text-align: center;
103    font-size: 10px;
104    line-height: 20px;
105}
106.qui-footerBasic .switchStyle {
107    color: #9ea3ab;
108}
109.qui-footerBasic .switchStyle span {
110    margin-left: 30px;
111}
112.qui-footerBasic .switchStyle a {
113    color: #444;  /*字体颜色为#444*/
114    text-decoration: none;  /*清除文本样式*/
115}
116.qui-asides {
117    position: absolute;
118    left: -200px;
119    top: 0;
120    height: 400px;
121    width: 200px;
122}
123.qui-aside {
124    position: fixed;
125    top: 0;
126    bottom: 0;
127    width: 200px;
128    overflow-y: scroll;  /*垂直方向内容溢出则出现滚动条*/
129    background-color: #2d3741;
130}
131.qui-asideNav {
132    /*上内边距为 80px，左右内边距都为 10px，下内边距为 10px*/
133    padding: 80px 10px 10px;
```

```
}
.qui-asideNav li {
    border-top: 1px solid #232d34;
    background-color: #36424b;
}
.qui-asideNav a {
    text-decoration: none;
    display: block;
    padding-left: 15px;
    font-size: 16px;
    line-height: 44px;
    color: #ced1d5;
}
.qui-asideNav a:hover {
    color: white;
    background-color: #2bab79;
}
```

【项目总结】

本项目的练习重点：

通过本项目的练习，读者可以掌握 HTML5 中的语义化标签的使用。本项目也涉及 CSS3 新增的属性，将在后面的单元中进行专题讲解。本项目不要求读者能够完成页面的编码，但是需要读者理解页面结构，探析实际开发中使用的样式设计。在重点代码处均有注释，读者能够读懂代码即可。

本项目的练习方法：

建议读者在学习时，先编写 HTML 文件的结构标签，然后探析 common.css 文件对通用样式进行设置，最后解读 main.css 文件对页面样式进行设置。

【思考题】

1. 请简述什么是响应式 Web 设计，并列举响应式 Web 设计需要应用哪些技术。

2. 请简述 HTML5 相比原来的版本有哪些更新，并列举 HTML5 常用的语义化标签（6 个以上）。

Responsive Web Design

2 Chapter

单元 2
文本类网页设计

网页的意义在于承载信息，而文字就是表达信息的一种方式，本单元将带领读者完成 3 个以文字为主题的网页制作。

【教学导航】

学习目标	1．掌握 HTML5 文本标签和 CSS3 文本外观属性 2．掌握 CSS3 中的@font-face 规则和多列布局模块 3．掌握 CSS3 中的字体图标 4．熟练使用 CSS 属性制作页面导航
教学方式	学习本单元之前，建议读者先了解【项目描述】，带着页面效果学习项目中涉及的知识点，然后动手进行编码实现项目效果。需要注意的是： 项目 2-1 先实现多列结构再调整字体样式； 项目 2-2 的难点在于字体图标的使用； 项目 2-3 的重点在于移动端的导航设计
重点知识	1．@font-face 规则 2．多列布局模块（Multi-column Layout Module） 3．Web 字体图标
关键词	font、多列、字体图标、导航

【项目 2-1】 电子杂志页面

【项目描述】

现在是电子化、智能化、网络化的时代，为了顺应时代的潮流，很多知名杂志都陆续出现了电子版产品，电子杂志版面里的文字模块、图文模块、影音模块等可以自由移动、自由组合，省去了纸质杂志更换版面的复杂程序。

本项目要完成一个简单、美观的电子杂志页面，主要以图文为主，页面效果如图 2-1 所示。

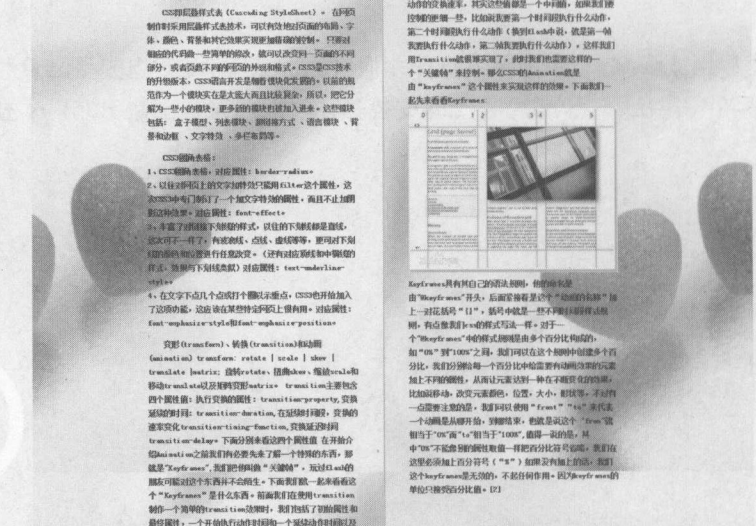


图2-1 电子杂志页面效果

【前导知识】

HTML5 中常用的文本标签

HTML5 中常用的文本标签如表 2-1 所示。

表 2-1　HTML5 文本标签

标　签	描　述
标题标签	HTML 中定义了 6 级标题，分别为 h1、h2、h3、h4、h5、h6，每级标题的字体大小依次递减，1 级标题字号最大，6 级标题字号最小
段落标签	<p>标签用于定义段落
 标签与<wbr>标签	 标签可插入一个简单的换行符，用来输入空行，而不是分割段落。 标签是空标签，在 XHTML 中，把结束标签放在开始标签中，也就是 ； <wbr>标签用于指明在文本中的何处适合添加换行符，其作用是建议浏览器在该标记处断行（只是建议而不是一定要在此处断行），最终是否断行根据整行文字长度而定。所有浏览器都支持 <wbr> 标签，除了 Internet Explorer
<details>标签与<summary>标签	<details>标签用于描述文档或文档某个部分的细节，目前只有 Chrome 浏览器支持<details>标签，可以与<summary>标签配合使用，<summary>标签用于定义该描述文档的标题
<bdi>标签	<bdi>标签用于设置一段文本，使其脱离其父标签的文本方向设置
<ruby>标签、<rt>标签与<rp>标签	<ruby>标签用于定义 ruby 注释（中文注音或字符），与<rt>标签一同使用； <rt>标签用于定义字符（中文注音或字符）的解释或发音； <rp>标签在 ruby 注释中使用，以定义不支持<ruby>标签的浏览器所显示的内容
<mark>标签	<mark>标签主要用来在视觉上向用户呈现那些需要突出显示或高亮显示的文字，典型应用是在搜索结果中高亮显示搜索关键字
<time>标签	<time>标签用于定义日期或时间，也可以两者同时定义
<meter>标签	<meter>标签用于定义度量衡，仅用于已知最大和最小值的度量
<progress>标签	<progress>标签用于定义任何类型任务的运行进度，可以使用<progress>标签显示 JavaScript 中时间函数的进程

CSS 的字体样式属性

CSS 的字体样式属性用于定义文本的字体系列、大小、风格等，CSS 常用的字体样式属性如表 2-2 所示。

表 2-2　CSS 常用的字体样式属性

属　性	允许取值	描　述
font-size：字号大小	1em、5em 等	em 表示相对于当前对象内文本的字体尺寸
	5px	px 表示像素，最为常用，推荐使用
font-family：字体	"微软雅黑"	网页中常用的字体有宋体、微软雅黑、黑体等
font-weight：字体粗细	normal	默认值，定义标准的字符
	bold	定义粗体字符
	bolder	定义更粗的字符
	lighter	定义更细的字符
	100~900（100 的整数倍）	定义由细到粗的字符，其中 400 等同于 normal，700 等同于 bold，值越大字体越粗

续表

属　性	允许取值	描　述
font-style：字体风格	normal	默认值，浏览器会显示标准的字体样式
	italic	浏览器会显示斜体的字体样式
	oblique	浏览器会显示倾斜的字体样式
word-wrap：长单词或 URL 自动换行	normal	只在允许的断字点处换行（浏览器保持默认处理）
	break-word	在长单词或 URL 地址内部进行换行

1. 字体属性合写

除以上常用的属性外，CSS 还支持字体的合写功能，font 属性用于对字体样式进行综合设置，其基本语法格式如下。

```
选择器{font: font-style || font-weight || font-size ||line-height || font-family;}
```

2. CSS3 的@font-face 规则

@font-face 是 CSS3 的新特性，用于定义服务器字体。通过@font-face 规则，开发者便可以使用用户计算机中未安装的字体。

@font-face 规则的语法格式如下。

```
@font-face {
        font-family: <YourWebFontName>;
        src: <source> [<format>][,<source> [<format>]]*;
        [font-weight: <weight>];
        [font-style: <style>];
      }
```

@font-face 规则的取值说明如下。

- YourWebFontName：自定义的字体名称，最好是使用下载的默认字体（如下载字体名称为 myFont，这里就填写为“myFont”），它将被引用到 Web 元素中的 font-family，如“font-family: "myFont"”。
- source：此值指的是自定义的字体存放路径，可以是相对路径也可以是绝对路径。
- format：此值指的是自定义的字体格式，主要用来帮助浏览器识别自定义的字体格式，其值主要有 truetype、opentype、truetype-aat、embedded-opentype、svg 等几种类型。
- weight 和 style：这两个值大家一定很熟悉，weight 定义字体是否为粗体，style 主要定义字体样式，如斜体。

@font-face 规则的具体用法如 demo2-1 所示。

demo2-1.html

```
1  <!DOCTYPE html>
2  <html>
3  <head lang="en">
4      <meta charset="UTF-8">
5      <title>@font-face 用法</title>
6  </head>
7  <style>
8  @font-face {
9      font-family: myFont;
```

```
    src:url('fonts/demo2-1/书法.ttf');
}
    div{
        font-family: myFont;
        font-size: 4em;
    }
</style>
<body>
<div>
    使用@font-face，应用漂亮的 Web 字体
</div>
</body>
</html>
```

用浏览器打开 demo2-1，页面效果如图 2-2 所示。

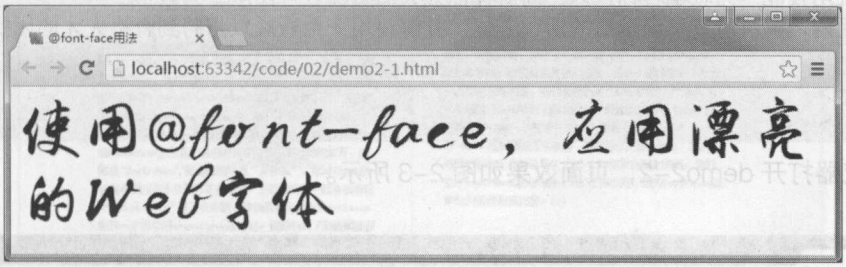


图2-2　demo2-1页面效果

在 demo2-1 中，通过@font-face 规则设置了自定义字体，font-family 属性表示为自定义字体取的名字，src 属性用来设置文件的路径，“书法.ttf”为字体文件，读者可以根据自己的喜好下载字体。需要注意的是，最后需要对页面文字进行字体设置，如代码第 12～15 行。

多列布局

在网页制作中，如果想让一个段落的内容像报刊、杂志中那样分栏显示，就需要用到 CSS3 中提出的多列布局模块（Multi-column Layout Module）。多列布局的具体用法如 demo2-2 所示。

demo2-2.html

```
<!DOCTYPE html>
<html>
<head lang="en">
    <meta charset="UTF-8">
    <title>多列布局</title>
</head>
<style>
p{
    width: 1500px;
    /*设置列数*/
    -webkit-column-count: 2;
    /*指定每列固定宽度，列的实际宽度和容器宽度有关*/
    -webkit-column-width: 450px;
```

```
    /*设置列与列之间的空隙*/
    -webkit-column-gap: 5em;
    /*设置列中间的分割线，与 border 的值类似*/
    -webkit-column-rule: 5px solid silver;
    font-size: 27px;
    margin: 0 auto;
}
</style>
<body>
<p>
        北京传智播客教育科技有限公司是一家专门致力于高素质软件开发人才培养的高科技公司。
    它依托程序员平台 csdn，整合了国内众多知名软件企业的资源，
    并邀请跨国公司和国内大中型企业架构师、系统分析师、企业培训师组成自己的团队。传智
    播客致力于为企业培养人才的培训理念，以“学员自学入门教程，
    通过基础考核后进行强化培训”为招生原则，以“针对企业需求，重视基础理论建设，强化高端
    应用技能”为教学目标，以“高薪保证强大的资深教育团队”为教学
    后盾，解决所有培训学员的后顾之忧，并解决用人企业难以招聘到合格人才的困扰。
</p>
</body>
</html>
```

用浏览器打开 demo2-2，页面效果如图 2-3 所示。

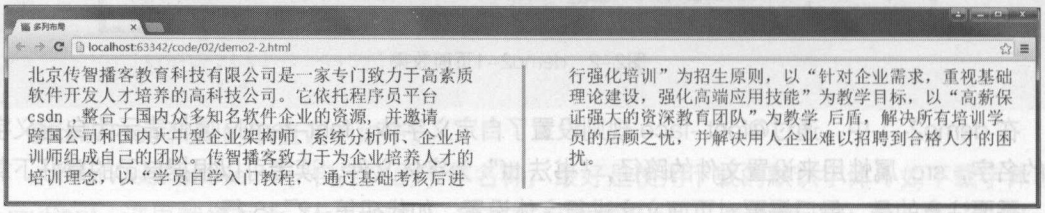


图2-3 demo2-2页面效果

在 demo2-2 中，使用了 CSS3 的多列布局模块的一些列属性来设置列数、每列宽度、列与列之间的间隙和分割线。CSS3 多列布局模块在一定程度上提高了 HTML 文档的表现力，在实际开发中应用广泛，如电子书、新闻页面等。

【项目分析】

有了前导知识作为铺垫，相信读者能够很好地完成电子杂志页面了。下面，我们对项目的构成进行分析。该页面的页面标注和页面结构如图 2-4 和图 2-5 所示。

如图 2-5 所示，该电子杂志页面由 1 个 标签嵌套多个<p>标签段落构成。该页面的实现细节具体分析如下。

（1）对 body 元素设置了一个背景图，作为杂志背景。

（2）标题部分可以使用标题标签<h4>，右侧文字插入的图片可以用<img>标签定义。

（3）使用多列布局的知识对网页文字进行布局，会出现一条灰色的线将文字分成两列。

（4）使用@font-face 规则定义自定义的字体，并控制段落文本的字号、粗细、颜色和首行缩进等样式。

图2-4　页面标注

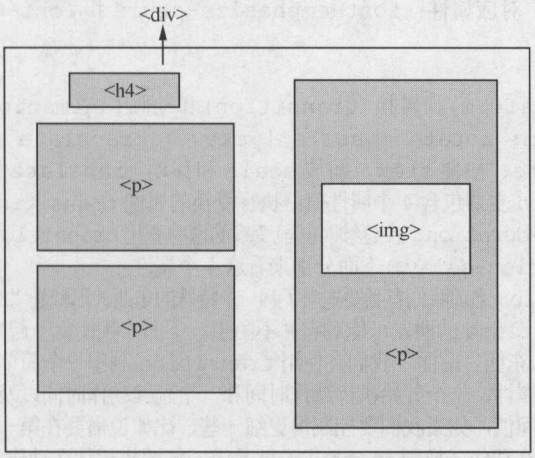


图2-5　页面结构

【代码实现】

了解了项目的结构和样式后，即可开始用代码来实现它。HTML 页面代码如 code\02\0201\0201.html 所示。

0201.html

```
1  <!DOCTYPE html>
2  <html lang="en">
```

```
<head>
    <meta charset="UTF-8">
    <title>CSS3 电子杂志页面</title>
    <link href="css/e-zine.css" type="text/css" rel="stylesheet">
</head>
<body>
<div class="wrapper">
    <h4>CSS3 简介</h4>
    <p>
        CSS 即层叠样式表（Cascading Style Sheet）。 在网页制作时采用层叠样式表技术，可以有效地对页面的布局、字体、颜色、背景和其他效果实现更加精确的控制。 只要对相应的代码做一些简单的修改，就可以改变同一页面的不同部分，或者页数不同的网页的外观和格式。CSS3 是 CSS 技术的升级版本，CSS3 的语言开发是朝着模块化发展的。以前的规范作为一个模块实在是太庞大了，而且比较复杂，所以，把它分解为一些小的模块，更多新的模块也被加入进来。这些模块包括盒子模型、列表模块、超链接方式、语言模块、背景和边框、文字特效、多栏布局等。
    </p>
     <p>
         CSS3 圆角表格：<br/>
            1. CSS3 圆角表格，对应属性：border-radius。<br/>
             2. 以往对网页上的文字加特效只能用 filter 属性，这次 CSS3 中专门制订了一个加文字特效的属性，而且不止加阴影这种效果。对应属性：font-effect。</br>
              3. 丰富了对链接下划线的样式，以往的下划线都是直线，这次可不一样了，有波浪线、点线、虚线等，更可对下划线的颜色和位置进行任意改变（还有对应顶线和中横线的样式，效果与下划线类似）。对应属性：text-underline-style。</br>
              4. 在文字下点几个点或打个圈以示重点，CSS3 也开始加入了这项功能，这应该在某些特定网页上很有用。对应属性：font-emphasize-style 和 font-emphasize-position。
    </p>
    <p>
        变形(transform)、转换(transition)和动画(animation)
        transform: rotate | scale | skew | translate |matrix;
        旋转 rotate、扭曲 skew、缩放 scale 和移动 translate 以及矩阵变形 matrix。
        transition 主要包含 4 个属性值：执行变换的属性 transition-property、变换延续的时间 transition-duration、在延续时间段变换的速率变化 transition-timing-function、变换延迟时间 transition-delay。下面分别来看这 4 个属性值。在开始介绍 Animation 之前我们有必要先来了解一个特殊的东西，那就是“Keyframes”，我们将其称为“关键帧”，玩过 flash 的朋友可能对它并不陌生。下面，我们就一起来看看这个“Keyframes”是什么东西。前面，我们在使用 transition 制作一个简单的 transition 效果时，涉及了初始属性和最终属性，一个开始执行动作时间和一个延续动作时间以及动作的变换速率，其实这些值都是一个中间值，如果我们要控制得更细一些，比如说需要在第一个时间段执行什么动作，在第二个时间段执行什么动作（换到 flash 中，就是第一帧要执行什么动作，第二帧要执行什么动作），此时我们用 Transition 就很难实现了，就需要这样的一个“关键帧”来控制。那么，CSS3 的 Animation 就是用“keyframes”这个属性来实现这样的效果。下面，我们一起先来看看 Keyframes。
        <img src="./images/pic.jpg" alt="CSS3" style="">
        Keyframes 具有自己的语法规则，其命名是以“@keyframes”开头，后面紧接着是这个“动画的名称”加上一对花括号“{}”，括号中就是一些不同时间段的样式规则，有点类似于 CSS 的样式写法。对于一个“@keyframes”中的样式规则是由多个百分比构成的，如“0%”到“100%”之间，我们可以在这个规则中创建多个百分比。我们分别在每一个百分比中给需要有动画效果的元素加上不同的属性，从而让元素达到一种在不断变化的效果，如移动、改变元素的颜色、位置、大小、形状等，不过有一点需要注意的是，我们可以使用“fromt”“to”来代表一个动画是从哪开始、到
```

```
53 哪结束，也就是说这个“fromt”就相当于“0%”，而“to”相当于“100%”，值得一提的是，其中“0%”不
54 能像别的属性取值一样把百分比符号省略，这里必须加上百分符号（“%”）如果没有加上，
55 则该 keyframes 是无效的，不起任何作用。因为 keyframes 的单位只接受百分比值。[2]
56     </p>
57 </div>
58 </body>
59 </html>
```

该项目的 CSS 样式代码如 code\02\ 0201\ css\e-zine.css 所示。

e-zine.css

```
1  /*第 2 单元 项目 2-1  电子杂志页面 */
2  body,  h4 {
3      margin: 0;  /*外边距为 0*/
4      padding: 0;  /*内边距为 0*/
5  }
6  /*给整个 body 设置一张背景图*/
7  body{
8      background-image: url(../images/bg.jpg) ;
9  }
10 @font-face {
11     font-family: myFont;
12     src:url('../fonts/简方叠体.TTF');
13 }
14 .wrapper {
15     width: 1024px;
16     padding: 30px;  /*上、下、左、右的内边距都是 30px; */
17     line-height: 1.75em;/ *行高为 1.75em*/
18     margin: 0 auto;/ *水平居中*/
19     font-family: myFont;
20     color: #333;
21     box-sizing: border-box;
22     -webkit-column-count: 2;/ *列数为 2*/
23     -webkit-column-width: 400px;/ *每列的宽度为 400px*/
24     -webkit-column-gap: 100px;/ *列与列之间的空隙为 100px*/
25     -webkit-column-rule: 3px solid #CCC;/ *列中间的分割线为#ccc 颜色的 3px 实线*/
26     text-indent:2em;/ *文本首行缩进 2 个文字*/
27 }
28 h4 {
29     font-size: 24px;
30     -webkit-column-span: all;/ *设置对象元素横跨所有列*/
31 }
32 p{
33     font-weight: bold;/ *字体为粗体*/
34 }
```

【项目总结】

本项目的练习重点：

本项目重点练习多列布局和 Web 字体的使用。

本项目的练习方法：

建议读者在编写 HTML 页面时，先编写页面结构，然后将文本和图片粘贴到对应的标签中。在编写 CSS 文件时，要先实现多列布局，再对字体、段落等细节进行调整。

本项目的注意事项：

该项目代码中，多列布局部分的代码加上了谷歌浏览器的-webkit 私有前缀，如果使用其他浏览器没有达到预定的效果，可以将前缀替换为对应的浏览器私有前缀。

【项目 2-2】 软文推广页面

【项目描述】

广告软文是一种网站推广的表现形式，其精美的外观和精简的内容信息很容易被用户所接受，从而达到很好的推广效果，本项目将带领读者实现一个软文推广页面。

页面效果如图 2-6 所示。

图2-6　　广告推广软文页面

【前导知识】

CSS 的文本外观属性

CSS 的文本外观属性用于设置颜色、字间距、字母间距、水平对齐、文本装饰、阴影等，常用的文本外观属性如表 2-3 所示。

表 2-3　文本外观属性

属性	允许取值	描　述
color： 文本颜色	red，green，blue	预定义的颜色值
	#FF0000，#FF6600， #29D794	十六进制颜色值，也是最常用的定义颜色的方式
	rgba（255,0,0,0.5）或 rgba （100%,0%,0%,0.5）	r：红色值；g：绿色值；b：蓝色值，r、g、b 的取值可以是正整数也可以是百分数。a：透明度，取值为 0~1
letter-spacing： 字间距	normal，0.5em，30px	用于定义字符与字符之间的空白，normal 为默认值，其属性值可为不同单位的数值，允许使用负值

续表

属性	允许取值	描　述
word-spacing： 单词间距	normal，0.5em，30px	用于增加或减少单词间的空白（即字间隔）。默认值为 normal，其属性值可为不同单位的数值，允许使用负值
line-height： 行间距	5px，3em，150%	用于定义行与行之间的距离，属性值单位有 3 种，分别为像素 px、相对值 em 和百分比%，实际工作中使用最多的是像素 px
text-transform： 英文文本转换	none	不转换（默认值）
	capitalize	首字母大写
	uppercase	全部字符转换为大写
	lowercase	全部字符转换为小写
text-decoration： 文本装饰	none	没有装饰（正常文本默认值）
	underline	设置文本下划线
	overline	设置文本上划线
	line-through	设置文本删除线
text-align： 水平对齐方式	left	左对齐（默认值）
	right	右对齐
	center	居中对齐
text-indent： 首行缩进	2em，50px，30%	用于设置首行文本的缩进，其属性值可为不同单位的数值、em 字符宽度的倍数或相对于浏览器窗口宽度的百分比%，允许使用负值，建议使用 em 作为设置单位
white-space： 空白符处理	normal	常规（默认值），文本中的空格、空行无效，满行（到达区域边界）后自动换行
	pre	预格式化，按文档的书写格式保留空格、空行原样显示
	nowrap	合并所有空白符为一个空白符，强制文本不能换行，除非遇到换行标记 。内容超出元素的边界也不换行，若超出浏览器页面则会自动增加滚动条
text-overflow： 标示对象内溢出文本	clip	修剪溢出文本，不显示省略标记“...”
	ellipsis	用省略标记“...”标示被修剪文本，省略标记插入的位置是最后一个字符。需要结合 overflow:hidden;使用

多学一招：text-shadow 为文本添加阴影效果

在 CSS 中，使用 text-shadow 属性可以为页面中的文本添加阴影效果，其基本语法格式如下。

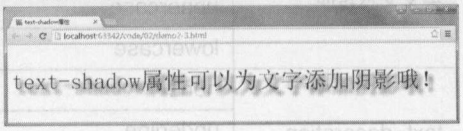
```
选择器{text-shadow:h-shadow v-shadow blur color;}
```

在上面的语法格式中，h-shadow 用于设置水平阴影的距离，v-shadow 用于设置垂直阴影的距离，blur 用于设置模糊半径，color 用于设置阴影颜色。text-shadow 属性的具体用法如 demo2-3 所示。

demo2-3.html

```
1  <! DOCTYPE html>
2  <html>
3  <head>
```

```
    <meta charset="utf-8">
    <title>text-shadow 属性</title>
    <style type="text/css">
        P{
            font-size: 50px;
            /*设置文字阴影的垂直距离、水平距离、模糊半径和颜色*/
            text-shadow:10px 10px 10px #2c41ff;
}
    </style>
</head>
<body>
<p>改变中国 IT 教育，我们正在行动...</p>
</body>
</html>
```



用浏览器打开 demo2-3，页面效果如图 2-7 所示。　　图2-7　demo2-3页面结果

CSS 的层叠性、继承性和重要性

1. CSS 的层叠性

如果在 HTML 文件中对于同一个标签可以有多个 CSS 样式存在，并且这多个 CSS 样式具有相同的权重值怎么办？CSS 的层叠性可以解决这个问题。

CSS 的层叠性是指在 HTML 文件中对于同一个标签元素可以有多个 CSS 样式存在，当有相同权重的样式存在时，会根据这些 CSS 样式的前后顺序来决定，处于最后面的 CSS 样式会被应用。

CSS 层叠性的示例代码如下。

```
<style>
p{color:red;}
p{color:green;}
</style>
<p >小红是一个胆小如鼠的女孩。</p>
```

由于 CSS 的层叠性，上面代码中的“小红是一个胆小如鼠的女孩”应显示为绿色。

2. CSS 的继承性

如果在 HTML 文件中一个标签嵌套另一个标签，两个标签中都有文字，要求显示为一种颜色，那么，是不是要分别进行设置呢？

其实，CSS 中的某些样式是具有继承性的，它允许样式不仅应用于某个特定 HTML 标签，而且应用于其后代。

CSS 继承性的示例代码如下。

```
<style>
p{color:pink;}
</style>
<p>小红是一个<span>胆小如鼠</span>的女孩。</p>
```

在上面代码中，“小红是一个胆小如鼠的女孩”应显示为粉色，“胆小如鼠”4 个字也为粉色的原因是<p>标签的颜色设置被其后代标签继承了。

3. CSS 的重要性

我们在做网页代码的时候，一些特殊情况下需要为某些样式设置具有最高权值，此时我们可以使用!important 来解决。

CSS 重要性的示例代码如下。

```
<style>
p{color:red!important;}
p{color:green;}
</style>
<p >小红是一个胆小如鼠的女孩。</p>
```

在上面代码中“小红是一个胆小如鼠的女孩”应显示为红色。

注意

!important 要写在分号“;”的前面。

CSS 的优先级

CSS 的优先级是指 CSS 样式在浏览器中被解析的权重不同。

1. 样式优先级

多重样式（Multiple Styles）：如果外部样式、内部样式和内联样式同时应用于同一个元素，就是使用多重样式的情况。

一般情况下，优先级如图 2-8 所示。

外部样式 < 内部样式 < 内联样式

图2-8　样式优先级

2. 选择器的优先级

CSS 选择器的优先级如图 2-9 所示。

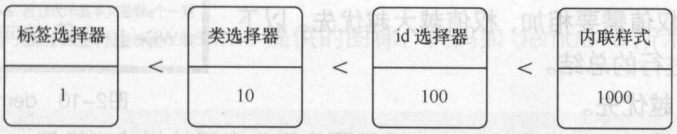


图2-9　选择器优先级

解释如下。

- 内联样式表的权值最高为 1000。
- id 选择器的权值为 100。
- class 类选择器的权值为 10。
- html 标签选择器的权值为 1。

需要注意的是，!important 的权值高于所有样式。接下来，我们一起来看一个通过选择器的权值进行比较的案例，如 demo2-4 所示。

demo2-4.html

```
1  <!DOCTYPE html>
2  <html>
3  <head lang="en">
4      <meta charset="UTF-8">
5      <title>选择器优先级</title>
6      <style type="text/css">
7          #redP p {
8              /* 权值 = 100+1=101 */
9              color:#F00;  /* 红色 */
```

```
        }
        #redP .red em {
            /* 权值 = 100+10+1=111 */
            color:#00F; /* 蓝色 */
        }
        #redP p span em {
            /* 权值 = 100+1+1+1=103 */
            color:#FF0;/*黄色*/
        }
    </style>
</head>
<body>
<div id="redP">
    <p class="red">
        第一个 p 标签文本显示为红色
        <span><em>em 标签文本显示为蓝色</em></span>
    </p>
    <p>第二个 p 标签文本显示为红色</p>
</div>
</body>
</html>
```

用浏览器打开 demo2-4，页面效果如图 2-10 所示。

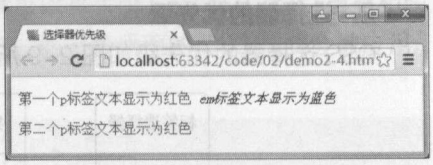


图2-10 demo2-4页面效果

通过 demo2-4 可以看出，选择器都有一个权值，使用复合选择器时，权值需要相加，权值越大越优先。以下是对 CSS 优先级进行的总结。

（1）权值越大越优先。

（2）当权值相等时，后出现的样式表设置要优于先出现的样式表设置。

（3）编写者的规则高于浏览者，即网页编写者设置的 CSS 样式的优先权高于浏览器所设置的默认样式。

（4）继承的 CSS 样式权值小于后来指定的 CSS 样式。

（5）在同一组属性设置中标有“!important”规则的优先级最大。

Web 字体图标——font-awesome 的应用

在传统的网页制作过程中，涉及图标的情况大多用图片进行处理，图片有优势也有不足。例如，使用图片会增加总文件的大小和很多额外的“http 请求”，增大服务器的负担，并且大量图片下载需要时，增加用户的等待时间，牺牲用户体验。另外，图片通常都是矢量图，在移动端高分辨率屏上会变得模糊，因此，有时候在“响应式设计”中需要使用图标的最佳解决方案就是不去使用图片，而字体通常是矢量的，所以就解决了图片的缺点，即图标字体化。

自定义图标字体是比较麻烦的，而 font-awesome 为用户准备了将近 500 个常用 icon 图标（并且还在不断更新），能大能小，能随便修改颜色，完全开源，且完全免费。

1. 下载

font-awesome 其实就是一个图标工具，当前的最新版本是 4.5.0。

首先，我们去“https://github.com/FortAwesome/Font-Awesome”地址进行下载，这个地址是一个面向开源及私有软件项目的托管平台，下载速度较官网快很多，解压之后，打开文件夹

目录如图 2-11 所示。

在图 2-11 中，我们只需关注“css”和“fonts”两个文件夹，“fonts”文件夹中有我们需要的字体文件，“css”文件夹中是该工具提供的 css 文件，将所有字体文件和 css 文件“font-awesome.min.css”拷贝到项目中，结构如图 2-12 所示。

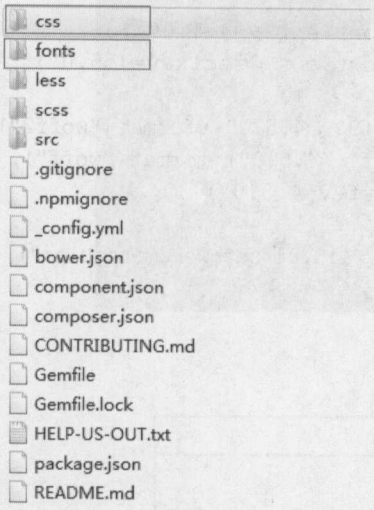


图2-11 文件夹目录

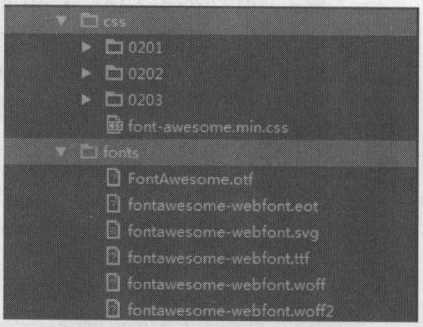


图2-12 项目中的文件目录

2. 应用字体

接下来介绍如何使用 font-awesome 提供的图标，代码如 demo2-5 所示。

demo2-5.html

```
<!DOCTYPE html>
<html>
<head lang="en">
    <meta charset="UTF-8">
    <title>字体图标应用</title>
  <link href="css/font-awesome.min.css" rel="stylesheet" type="text/css" />
</head>
<body>
<i class="fa fa-comments" style="font-size: 2em"></i>
</body>
</html>
```

用浏览器打开 demo2-5，页面效果如图 2-13 所示。

图2-13 demo2-5页面效果

这是 font-awesome 官方提供的图标，如果想使用其他图标可以去地址“http://fontawesome.io/icons/”进行查看，这是 fontawesome 提供的图标库，在这里可以找到使用每个图标需要引用的 class 值。除了上面引用字体图标的方法外，开发者还可以自定义引用字体图标的类名，如 demo2-6 所示。

demo2-6.html

```
<!DOCTYPE html>
<html>
```

```
<head lang="en">
    <meta charset="UTF-8">
    <title>字体图标应用</title>
    <style>
        @font-face {
            font-family: 'FontAwesome';
            src: url("fonts/fontawesome-webfont.eot?v=4.5.0");
            src: url("fonts/fontawesome-webfont.eot?#iefix&v=4.5.0")
            format("embedded-opentype"),
            url("fonts/fontawesome-webfont.woff2?v=4.5.0") format("woff2"),
            url("fonts/fontawesome-webfont.woff?v=4.5.0") format("woff"),
            url("fonts/fontawesome-webfont.ttf?v=4.5.0")
            format("truetype"),
            url("fonts/fontawesome-webfont.svg?v=4.5.0#fontawesomeregular")
            format("svg");
            font-weight: normal;
            font-style: normal;
        }
        /*设置字体编码*/
        .fa-weixin:before {
            content: "\f1d7";
        }
        .fa {
            color: green;
            font:  normal normal 50px FontAwesome;
            -webkit-font-smoothing: antialiased;
        }
    </style>
</head>
<body>
<i class="fa fa-weixin"></i>
<h5>微信公共平台开发</h5>
</body>
</html>
```

用浏览器打开 demo2-6，页面效果如图 2-14 所示。

图2-14　demo2-6页面效果

demo2-6 代码中的“fa-weixin”是自定义的，这里将第 23 行的 content 属性值设置为微信图标对应的编码，这个编码是使用 UI 设计师提供给前端人员的 svg 矢量图生成的，也可以在网上通过第三方网站获得，这里作为了解即可。

其实在 font-awesome 字体文件中，每个图标都有其对应的编码，font-awesome 提供的“font-awesome.min.css”文件中的代码与 demo2-6 中的代码相似，相当于替我们完成了这部分编码，所以我们使用字体图标时只需引用对应的 class 值，读者可以查看源码进行了解。

【项目分析】

有了前导知识作为铺垫，接下来我们来试着完成这个广告推广软文页面吧！

该页面的页面标注和页面结构如图 2-15 和图 2-16 所示。

图2-15　页面标注

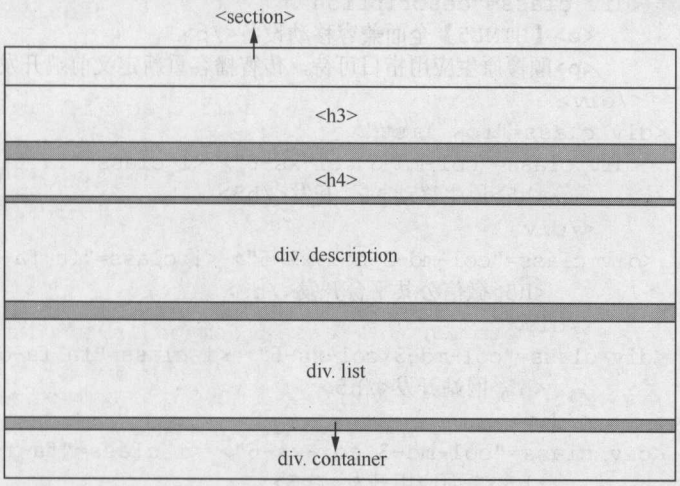


图2-16　页面结构

如图 2-16 所示，该软文推广页面由<section>标签嵌套多个 标签构成，广告中的文字分布在这些 标签中。

该页面的实现细节具体分析如下。

（1）最上面两个标题分别使用<h3>和<h4>标签，div.description 中的描述信息使用<p>标签来完成，字体图标使用<i>标签引用对应的 class 值来完成。

（2）为<section>标签设置蓝色的背景图。

（3）根据需求用 标签和 CSS 样式来控制文字的排列。

（4）在<h3>标签中应用了名称为“PinyonScript-Regular.ttf”的字体。

（5）<h3>和<h4>标签中的文字分别添加了灰色阴影。

【代码实现】

HTML 页面代码如 code\02\0202\0202.html 所示。

0202.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <link href="css/font-awesome.min.css" rel="stylesheet">
    <link href="css/advertising.css" rel="stylesheet" type="text/css" />
    <link rel="shortcut icon" href="favicon.ico">
    <title>广告推广软文页面</title>
</head>
<body>
    <section id="mobile" class="cover-section">
        <div class="section-inner">
            <div class="container">
                <h3>One Web, Any Device</h3>
                <h4>万物互联，极致体验</h4>
                <div class="description">
                    <p>【HTML5】全面兼容移动设备</p>
                    <p>颠覆原生应用指日可待，传智播客重新定义前端开发</p>
                </div>
              <div class="row list">
                <div class="col-md-3 col-xs-6"> <i class="fa fa-apple"></i>
                        <h5>原生移动 APP 开发</h5>
                    </div>
               <div class="col-md-3 col-xs-6"> <i class="fa fa-weixin"></i>
                        <h5>微信公共平台开发</h5>
                    </div>
              <div class="col-md-3 col-xs-6"> <i class="fa fa-desktop"></i>
                        <h5>网站开发</h5>
                    </div>
              <div class="col-md-3 col-xs-6"> <i class="fa fa-laptop"></i>
                        <h5>桌面应用开发</h5>
                    </div>
                </div>
            </div>
        </div>
    </section>
</body>
</html>
```

该项目的 CSS 样式代码如 code\02\0202\css\advertising.css 所示。

advertising.css

```
/*第 2 单元 项目 2-2  广告推广软文页面 */
html,body {
  width: 100%;
  height: 100%;
}
html{
  overflow: hidden;
```

```
}
body {
  font-family: 'Microsoft Yahei';
  color: #ffffff;
  /*新增的，去 body 的默认边距*/
  padding: 0;
  margin: 0;
}
 h3, h4, h5{
  margin-top: 0;
  margin-bottom: .5rem;
}
p {
  margin-top: 0;
  margin-bottom: 1rem;
}
div {
    display: block;
}
@font-face {
  font-family: 'Pinyon Script';
  src: url("../fonts/PinyonScript-Regular.ttf") ;
  font-weight: normal;
  font-style: normal;
}
.fa {
  display: inline-block;/*把行元素转换为行内块元素*/
  font: normal normal normal 14px / 1 FontAwesome;
  font-size: inherit;
  text-rendering: auto;/*设置文本渲染引擎工作时如何优化显示文本， 默认值：auto*/
  -webkit-font-smoothing: antialiased;
}
#mobile {
  background-image: url("../images/section_mobile_bg.jpg");
  text-align: center;
}
#mobile h3 {
  font-size: 6rem;
  font-family: 'Pinyon Script';
  font-weight: 600;
/*设置文字阴影：向右 5px，向下 5px，模糊程度 0.3rem，颜色为#62819d*/
  text-shadow: 5px 5px 0.3rem #62819d;  margin-bottom: 2rem;
}
#mobile h4 {
  font-size: 3rem;
  text-shadow: 10px 10px 0.2rem #62819d;
}
.cover-section {
  background-size: cover;/*背景平铺*/
  background-repeat: no-repeat;/*背景不重复*/
```

```
  background-position: center;
  display: table;/*此元素会作为块级表格来显示（类似 <table>）*/
  height: 100%;
  width: 100%;
  position: relative;
}
 .section-inner {
  display: table-cell;/*此元素会作为一个表格单元格显示（类似 <td> 和 <th>）*/
  width: 100%;
  vertical-align: middle;/*垂直居中*/
}
#mobile .description {
  margin-top: 3rem;
}
#mobile .description > p {
  font-size: 1.5rem;
}
.row {
  margin-left: -0.9375rem;
  margin-right: -0.9375rem;
}
.row:before, .row:after {
  content: " ";
  display: table;
}
.row:after {
  clear: both;/*清除左右浮动*/
}
#mobile .list {
  text-align: center;
  margin: 3rem 0 1.5rem;/*外边距：上边距 3rem，左右边距 0rem，下边距 1.5rem*/
}
#mobile .list i {
  font-size: 3rem;
  padding: 2rem;
}
#mobile .list h5 {
  font-size: 1.3rem;
  font-weight: normal;
}
.col-xs-6,.col-md-3{
  position: relative;
  min-height: 1px;
  padding-left: 0.9375rem;
  padding-right: 0.9375rem;
  /*新增的*/
  float: left;
  box-sizing: border-box;
}
.col-xs-6 {
```

```
108  width: 50%;
109}
110.col-md-3 {
111  float: left;
112  width: 25%;
113}
```

【项目总结】

本项目的练习重点：

通过本项目的练习，读者要熟练掌握字体图标的使用。

本项目的练习方法：

建议读者先进行结构代码的编写，再进行样式的添加。

本项目的注意事项：

（1）该项目在字体图标部分所应用的文件可以在网上进行下载，本教材的源码中也会提供，不要忘记在 HTML 页面引入“css/font-awesome.min.css”，这个文件中有部分代码需要在该路径下。

（2）CSS 文件中的路径需要修改成用户本地的路径，也可以按照代码中的位置存放文件，制作过程中注意字体图标的应用和各种文字的设置。

【项目 2-3】 手机邮箱导航页面

【项目描述】

现在大部分企业都在使用企业邮箱，因为它安全、高效，不仅在 PC 端提供支持，而且在移动端也提供了更自由的办公体验，这就是手机邮箱。本项目要带领读者完成一个手机邮箱的导航页面，如图 2-17 所示。

图2-17　手机邮箱导航栏

【前导知识】

CSS 链接属性

在实际开发中，网页中的链接有 4 种状态，具体如下。

- a:link：超链接的初始状态。
- a:hover：把鼠标放上去时悬停的状态。
- a:active：是鼠标单击时的状态。
- a:visited：是访问过后的状态。

我们要为上面 4 种不同状态的链接设置不同的样式来区分它们的状态。这 4 种状态的排序有一个有趣的规则——LoVe HAte 原则，即按照 link、visited、hover、active 的顺序进行设置。

改变超链接样式的示例代码如下。

```
/*对全站有链接的文字设置颜色样式为 color:#333; 并无下划线*/
a{color:#333;text-decoration:none; }
/*对鼠标放到超链接上的文字设置颜色样式为 color:#CC3300;
```

```
且文字链接加下划线 text-decoration:underline; */
a:hover {color:#CC3300;text-decoration:underline;}
```

CSS 导航栏

对于任何一个网站来说，拥有一个易用的导航栏都是非常重要的，可以大大提高用户体验。导航栏其实就是一个超链接的列表，所以使用无序列表<ul>标签来实现它再合适不过了。导航栏的示例代码如 demo2-7 所示。

demo2-7.html

```
<!DOCTYPE html>
<html>
<head lang="en">
    <meta charset="UTF-8">
    <title>导航栏</title>
    <style>
        ul{
            list-style-type: none;
            margin: 0;
            padding: 0;
        }
        /*对全站有链接的文字设置颜色，并无下划线*/
        a{color:#333;text-decoration:none; }
        /*对鼠标放到超链接上的文字设置颜色，且文字链接加下划线*/
        a:hover {color: #ff898e;text-decoration:underline;}
        li{
            float: left;/*li 元素浮动的水平导航效果*/
        }
    </style>
</head>
<body>
<ul>
    <li><a href="http://www.baidu.com/">百度</a></li>
    <li><a href="http://www.sina.com/">|新浪</a></li>
    <li><a href="http://www.itcast.cn/">|传智播客</a></li>
</ul>
</body>
</html>
```

用浏览器打开 demo2-7，页面效果如图 2-18 所示。

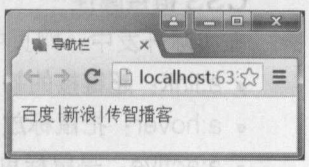


图2-18 demo2-7页面效果

导航菜单的各种效果如垂直、水平都可以根据样式来控制。在 demo2-7 中，使用 li 元素浮动的水平效果；除此之外，使用行内列表项 li{display:inline}也可以实现同样的效果。读者可以自行尝试。

【项目分析】

有了前导知识作为铺垫，接下来我们一起来分析手机邮箱导航页面吧！该页面的页面标注和页面结构如图 2-19 和图 2-20 所示。

如图 2-20 所示，该导航菜单页面由一个大的 标签嵌套类名为 menu 的 标签构成，页面的主体部分都包含在 div.menu 中。

图2-19　页面标注

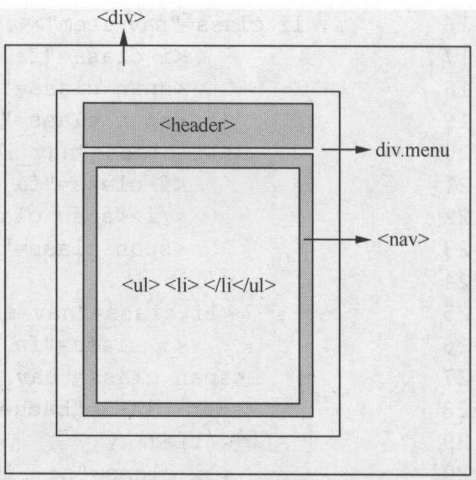


图2-20　页面结构

该页面的实现细节具体分析如下。

（1）在 div.menu 中分为<header>标签部分和<nav>导航部分，导航菜单由<ul>嵌套<a>标签等构成。

（2）设置页面所有元素的基线同父元素的基线对齐，调整各个元素的位置。

（3）每个菜单左侧图标可以使用 font-awesome 图标库提供的图标，并为每个图标设置不同的颜色。

（4）统一设置每个菜单文字也就是<a>标签的样式，同时应用 a:hover 为鼠标悬停添加特定样式。

（5）每个菜单右侧带样式的文字用 CSS+标签进行控制。

【代码实现】

对项目的结构和样式有所了解后，即可开始编写代码来实现它。HTML 页面代码如 code\02\0203\0203.html 所示。

0203.html

```
<!DOCTYPE html>
<html>
<head lang="en">
    <meta charset="UTF-8">
    <title>手机邮箱导航</title>
    <link href="css/font-awesome.min.css" rel="stylesheet">
    <link rel='stylesheet' href='css/email.css'></head>
<body>
<div class="l-main">
    <div class="menu">
        <header class="menu_header">
            <h1 class="menu_header-title">我的邮箱</h1>
        </header>
        <nav class="menu_body">
            <ul class="nav">
```

```
        <li class="nav_item"><a href="#" class="nav_item-link is-active">
                <i class="fa fa-envelope nav_item-icon"></i>
                <span class="nav_item-text">收件箱</span>
                <span class="badge badge--warning">32</span> </a></li>
            <li class="nav_item"><a href="#" class="nav_item-link">
                <i class="fa fa-flag nav_item-icon">
                </i><span class="nav_item-text">红旗邮件</span>
                <span class="badge">7</span></a>
            </li>
            <li class="nav_item"><a href="#" class="nav_item-link">
                <i class="fa fa-space-shuttle nav_item-icon"></i>
             <span class="nav_item-text">已发送</span>
            <span class="badge">0/17</span></a>
             </li>
             <li class="nav_item"><a href="#" class="nav_item-link">
             <i class="fa fa-archive nav_item-icon">/i>
             <span class="nav_item-text">草稿箱</span>
             <span class="badge">2</span></a>
             </li>
             <li class="nav_item"><a href="#" class="nav_item-link">
            <i class="fa fa-trash nav_item-icon"></i>
            <span class="nav_item-text">已删除</span>
            <span class="badge">5</span></a>
             </li>
             <li class="nav_item"><a href="#" class="nav_item-link">
              <span class="nav_item-text">所有邮件</span> </a></li>
          </ul>
       </nav>
    </div>
</div>
</body>
</html>
```

该项目的 CSS 样式代码如 code\02 \0203\css \email.css 所示。

email.css

```
/*第 2 单元 项目 2-3  手机邮箱导航页面 */
html,body,div,span,h1,a,ul,li,nav {
    margin:0;
    padding:0;
    border:0;
    font-size:100%;
    font:inherit;
    vertical-align:baseline;/*元素的基线与父元素的基线对齐*/
}
body
{
   line-height:1em;
   line-height:1em;
   background: #F4F4F4;
   font-family: Arial, sans-serif;
```

```
    font-size: 14px;
    font-weight: lighter;
}
.l-main
{
    width: 530px;
    margin: 0 auto;
}
.menu
{
    width: 250px;
    margin: 40px;
    background: #fff;
/*盒子阴影：水平方向为 0px，垂直方向向下为 1px，模糊程度为 4px，黑色透明度为 30%*/
    box-shadow: 0 1px 4px rgba(0, 0, 0, 0.3);
    border-radius: 5px;/*边框圆角 5px*/
    float: left;
}
.menu_header
{
    background: #4B4F55;
    border-bottom: 1px solid #353A40;/*设置元素下边框为 1px、颜色为#353A40 的实线*/
    border-radius: 5px 5px 0 0;/*边框圆角：左上角、右上角为 5px，右下角、左下角为 0px*/
}
.menu_header-title
{
    color: #fff;
    padding: 15px;
/*文本阴影：垂直方向为 0px，水平方向向右为 1px，黑色透明度为 40%*/
    text-shadow: 0 1px 0 rgba(0, 0, 0, 0.4); }
.menu_body
{
    border-radius: 0 0 5px 5px;/*边框圆角：左上角、右上角为 0px，右下角、左下角为 5px*/
}
.nav
{
    list-style: none;
}
.nav_item
{
    position: relative;
}
.nav_item-link
{
    padding: 10px 15px;/*内边距：上下为 10px，左右为 1px*/
    text-decoration: none;/*文本样式：无*/
    color: #8B8E93;
    display: block;/*把行元素强制转换为块元素*/
    border-bottom: 1px solid #F0F0F0;
}
```

```
.nav_item-link:hover
{
    background: #f0f0f0;/*鼠标悬浮在该元素时，背景颜色变为#f0f0f0*/
}
/*设置当鼠标单击该元素时的背景颜色、文字颜色和下边框颜色，盒子内阴影：水平方向向右为 1px，颜色为#7A828D*/
.nav_item-link.is-active
{
    background: #6E757F;
    color: #fff;
    border-bottom-color: #4B4F55;
    box-shadow: 0 1px 0 #7A828D inset;
}
.nav_item-link.is-active:after
{
    content: '';
    display: block;
/*绝对定位，父元素要加 position: relative;记住“子绝父相”原则*/
    position: absolute;
    top: 50%;/*距父元素上边 50%*/
    right: -6px;/*距父元素左边-6px*/
    margin-top: -6px;/*顶部外边距为-6px*/
    border-top: 6px solid transparent;/*上边框：6px 的透明实线*/
    border-bottom: 6px solid transparent;
    border-left: 6px solid #6E757F;
}
.nav_item-link.is-active
{
    color: #fff;/*当鼠标单击该元素时，文字颜色为白色*/
}
.nav_item:last-child .nav_item-link
{
    border-bottom: none;/*底边框：无*/
}
.nav_item-icon
{
    width: 20px;
    text-align: center;/*文本水平居中*/
    font-size: 18px;
    margin-right: 10px;
}
.badge
{
    font-size: 12px;
    padding: 2px 8px;
    border: 1px solid #D1D1D1;
    border-radius: 10px;
    position: absolute;
    top: 10px;
    right: 15px;
```

```
115}
116.badge-warning
117{
118    background: #ED373F;
119    border-color: #ED373F;
120}
121.fa-flag{
122    color: red;
123}
124.fa-space-shuttle{
125    color: #508049;
126}
127.fa-archive{
128    color: orange;
129}
130.fa-envelope{
131    color: #35b3ff;
132}
```

【项目总结】

本项目的练习重点：

通过本项目的练习，读者能够熟练掌握 CSS 中的链接属性，并且学会如何制作导航栏。

本项目的练习方法：

建议读者先进行导航结构编写，再进行样式添加，最后细化链接的状态样式。

本项目的注意事项：

由于不同的浏览器默认的样式不同，email.css 文件的 2~9 行用来覆盖浏览器的默认样式，这样就避免了默认样式对网页样式的影响。

【思考题】

1. 请简要概括什么是 CSS 的层叠性、继承性和重要性。
2. 请简述什么是 CSS 的优先级，并做出总结。

Responsive Web Design

3 Chapter

单元 3
图文展示网页设计

为追求更为直观的浏览与交互体验，用户对网站的美观性和交互性的要求越来越高。上一单元，我们练习了文本类的网站制作，本单元将针对图文信息展示的页面进行练习，制作出更为美观和具有动态效果的网站。

【教学导航】

学习目标	1. 掌握 HTML 图像标签的使用 2. 掌握 CSS3 的背景设置、阴影和渐变 3. 掌握 CSS3 的圆角边框、过渡和变形 4. 掌握使用 CSS3 实现动画效果 5. 熟练掌握精灵技术
教学方式	本单元有很多 CSS 实现的动画效果，建议先编写网页结构，分块实现 CSS 效果。编写样式的过程中要着重于本单元重点知识实现的动画效果。 项目 3-1 中鼠标悬停的效果； 项目 3-2 中圆角按钮和商品图片的鼠标悬浮动画； 项目 3-3 中桃子晃动的效果
重点知识	1. CSS3 圆角边框、过渡和变形 2. CSS3 动画
关键词	img、figure、background、shadow、border-radius、CSS3 transitions、CSS3 transform、CSS3 animations、CSS Sprites

【项目 3-1】　黑马书城

【项目描述】

现如今，网购已经成为大多数人生活中必不可少的一部分，它使得人们足不出户就能买到自己心仪的商品，大大压缩了时间成本。购物网站通常会以图文混排的方式展示商品信息。本项目将带领读者实现一个名为“黑马书城”的网上书城页面，如图 3-1 所示。

图3-1　黑马书城页面

当鼠标悬停在某个商品上时，会出现商品的价格等信息，如图 3-2 所示。

【前导知识】

HTML5 常用图像标签

在网页制作过程中，为了使页面变得更加生动活泼，插入图像是必不可少的。接下来，我们分别介绍 HTML5 中常用的几种图像标签，具体如下。

1. <img>标签

<img>标签用于定义网页中的图像，语法格式如下。

```
<img src="图片路径" alt="图片无法显示时显示的文字" />
```

其中，src 属性和 alt 属性是<img>标签必需的属性。

2. <figure>标签和<figcaption>标签

figure 的中文意思是插图，当需要在网页中添加一个插图时，就可以使用<figure>标签来实现。<figcaption>标签是嵌套在<figure>标签中使用的，用于指定图片的标注。例如，插图是一只小鸡，就可以使用<figcaption>来标注“一只小鸡”，具体用法如 demo3-1 所示。

demo3-1.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>HTML 常用的图像标签</title>
</head>
<body>
<figure>
    <figcaption>一只呆萌的小鸡</figcaption>
    <p>拍摄者：papi 拍摄时间：2016 年 2 月</p>
    <img src="images/demo3-1/chicken.png" alt="抱歉，您的图片不能显示。">
</figure>
</body>
</html>
```

用浏览器打开 demo3-1，页面效果如图 3-3 所示。

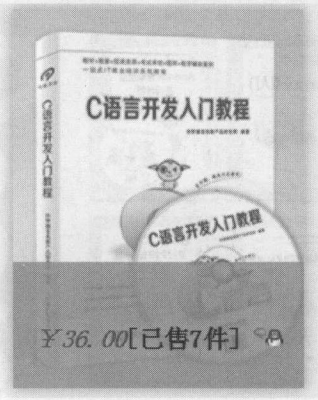


图3-2　鼠标悬停效果

图3-3　demo3-1页面效果

CSS 背景设置

在第 2 单元的项目中，曾经用到过设置网页的背景图的知识，其实这也是 CSS 背景设置知识的一部分。CSS 背景指的就是 CSS 对对象设置背景属性，如通过 CSS 设置背景的各种样式。

CSS 用于背景设置的常用属性如表 3-1 所示。

表 3-1　背景设置的常用属性

属性名	属性描述	允许取值	取值说明
background-color	设置背景色	red，green，blue	预定义的颜色值
		#FF0000，#FF6600，#29D794	十六进制颜色值，也是最常用的定义颜色的方式
		rgba(255,0,0,0.5) 或 rgba(100%,0%,0%,0.5)	r：红色值；g：绿色值；b：蓝色值，r、g、b 的取值可以是正整数也可以是百分数。a：透明度，取值为 0~1
background-image	设置图片背景	url (url)	直接引用图片地址来设置图片作为对象背景
background-repeat	设置背景平铺重复方向	repeat	背景图像在纵向和横向上平铺
		no-repeat	背景图像不平铺
		repeat-x	背景图像在横向上平铺
		repeat-y	背景图像在纵向上平铺
background-attachment	设置或检索背景图像是随对象内容滚动还是固定的	scroll	背景图像随对象内容滚动
		fixed	背景图像固定
background-position	设置或检索对象的背景图像位置，语法为 length|length 或者 position|position	35%、80%或 35% 2.5cm 或 top right	length：百分数 | 由浮点数字和单位标识符组成的长度值。position：top | center | bottom | left | center | right
background-size	规定背景图像的尺寸	length	第一个值设置宽度，第二个值设置高度。一个值时，第二个值会被设置为“auto”
		percentage	以父元素的百分比来设置背景图像的宽度和高度，用法同上
		cover	背景图完全覆盖背景区域
		contain	宽和高完全适应内容区域

除以上常用的背景属性外，CSS 支持背景属性的合写功能，也称为复合背景属性，用 background 进行设置，其基本语法格式如下。

```
选择器{background : background-color || background-image || background-repeat
|| background-attachment || background-position}
```

如使用该复合属性定义其单个参数，则其他参数的默认值将无条件覆盖各自对应的单个属性设置。默认值为 transparent none repeat scroll 0% 0%。尽管该属性不可继承，但如果未指定，其父对象的背景颜色和背景图将在对象下面显示。对应的脚本特性为 background。

CSS 阴影和渐变

1. 盒阴影

在单元 2 曾经讲解过 text-shadow 的使用方法，CSS3 的 box-shadow 有点类似于 text-shadow，不同的是 text-shadow 是对象的文本设置阴影，而 box-shadow 是给对象实现图层阴

影效果，其基本语法格式如下。

```
对象选择器 {box-shadow:投影方式|| x 轴偏移量|| y 轴偏移量 ||阴影模糊半径 ||阴影扩展半径 ||阴影颜色}
```

box-shadow 属性至多有 6 个参数设置，取值说明如表 3-2 所示。

表 3-2 box-shadow 属性参数说明

参数类型	取值说明
投影方式	此参数是一个可选值，如果不设值，其默认的投影方式是外阴影，设置阴影类型为“inset”时，其投影就是内阴影
x 轴偏移量	即阴影的水平偏移量，其值可以是正负值，如果值为正值，则阴影在对象的右边；反之，如果值为负值，则阴影在对象的左边
y 轴偏移量	即阴影的垂直偏移量，其值也可以是正负值，如果值为正值，则阴影在对象的底部；反之，如果值为负值，则阴影在对象的顶部
阴影模糊半径	此参数为可选，但其值只能为正值，如果值为 0，则表示阴影不具有模糊效果，其值越大阴影的边缘就越模糊
阴影扩展半径	此参数为可选，其值可以是正负值，如果值为正值，则整个阴影都延展扩大；反之，如果值为负值，则阴影缩小
阴影颜色	此参数为可选，如果不设定任何颜色，则浏览器会取默认色，但各浏览器的默认颜色不一样，特别是在 webkit 内核下的 safari 和 chrome 浏览器将无色，也就是透明，建议不要省略此参数

box-shadow 对盒子对象和图片对象都可以实现阴影效果，如 demo3-2 所示。

demo3-2.html

```
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>对象阴影</title>
    <link href="images/style.css" rel="stylesheet" type="text/css" />
    <style>
        .box {
           box-shadow:   7px 4px 10px #000 inset ;
             width:300px;
             height:80px;
        }
        .box1 img {
           box-shadow:#000 7px 4px 10px ;
        }
    </style>
</head>
<body>
<h3>盒子对象阴影测试</h3>
<div class="box">DIV 盒子内阴影</div>
<h3>图片对象阴影测试</h3>
<div class="box1"><img src="images/demo3-2/boxshadow.png" /></div>
</body>
</html>
```

用浏览器打开 demo3-2，页面效果如图 3-4 所示。

在 demo3-2 中分别设置 div 对象内阴影效果和图片外阴影效果，需要注意的是，box-shadow 的参数值的顺序不是固定的，几个像素值需要连在一起，投影方式写在第一个或者最后一个设置。

图3-4　demo3-2页面效果

2. CSS3 渐变

渐变是两种或多种颜色之间的平滑过渡，渐变背景一直以来在 Web 页面中都是一种常见的视觉元素。在 CSS3 以前，必须使用图像来实现这些效果。CSS3 的渐变属性主要包括线性渐变、径向渐变和重复渐变，下面重点讲解一下线性渐变和径向渐变。

（1）线性渐变

CSS3 中的线性渐变通过“background-image:linear-gradient（参数值）;”来设置，其基本语法格式如下。

```
background-image:linear-gradient([ <angle> | <side- or-corner>,] color stop, color stop[, color stop] *);
```

其中，[]中的参数为可选值。linear-gradient 的参数取值说明如表 3-3 所示。

表 3-3　linear-gradient 属性的参数说明

参数类型	取值说明
<angle>	表示渐变的角度，角度数的取值范围是 0～360deg。这个角度是以圆心为起点的角度，并以这个角度为发散方向进行渐变
<side-or-corner>	通过关键词来确定渐变的方向。默认值为 top（从上向下），取值范围是 [left,right,top, bottom,center,top right,top left,bottom left,bottom right,left center,right center]。注意：IE10 只能取[left,top]，Chrome 则没有[center,left center,right center]
color stop	用于设置颜色边界，color 为边界的颜色，stop 为该边界的位置，stop 的值为像素数值或百分比数值，若为百分比且小于 0%或大于 100%则表示该边界位于可视区域外。两个 color stop 之间的区域为颜色过渡区

线性渐变的具体使用方式如 demo3-3 所示。

demo3-3.html

```
<!DOCTYPE html>
<html>
<head lang="en">
    <meta charset="UTF-8">
    <title>CSS3 线性渐变</title>
</head>
<style type="text/css">
    .rainbow-linear-gradient{
        width: 460px;
        height: 160px;
        background-image: -webkit-linear-gradient(left,#E50743 0%, #F9870F 15%,
            #E8ED30 30%, #3FA62E 45%,#3BB4D7 60%,#2F4D9E 75%,#71378A 80%);
    }
</style>
<body>
```

```
15 <div class="rainbow-linear-gradient"></div>
16 </body>
17 </html>
```

用浏览器打开 demo3-3，页面效果如图 3-5 所示。

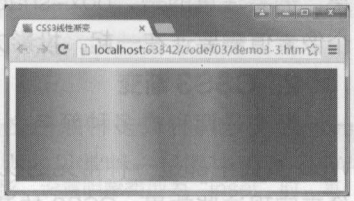


图3-5 demo3-3页面效果

在 demo3-3 中实现了一个七色彩虹的效果，每个颜色值后面的百分数表示该色标的位置比例。需要注意的是，关键词还可以用 to top（从下向上）、to top left（右下角到左上角）来表示。

（2）径向渐变

CSS3 中的径向渐变通过“background-image: radial-gradient（参数值）;”来设置，其基本语法格式如下。

```
background-image: radial-gradient(圆心坐标, 渐变形状 渐变大小, color stop, color stop[, color stop] *);
```

radial -gradient 的参数取值说明如表 3-4 所示。

表 3-4 radial -gradient 属性的参数说明

参数类型	取值说明	
圆心坐标	用于设置放射圆形的坐标，可设置为形如 10px 20px 的 x-offset y-offset，或使用预设值 center（默认值）	
渐变形状	circle	圆形
	ellipse	椭圆形，默认值
渐变大小	closest-side 或 contain	以距离圆心最近的边的距离作为渐变半径
	closest-corner	以距离圆心最近的角的距离作为渐变半径
	farthest-side	以距离圆心最远的边的距离作为渐变半径
	farthest-corner 或 cove	以距离圆心最远的角的距离作为渐变半径

径向渐变的具体使用方式如 demo3-4 所示。

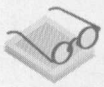
demo3-4.html

```
1  <!DOCTYPE html>
2  <html>
3  <head lang="en">
4      <meta charset="UTF-8">
5      <title>CSS3 径向渐变</title>
6      <style type="text/css">
7          .rainbow-radial-gradient{
8              width: 300px;
9              height: 300px;
10             background-image: -webkit-radial-gradient(100px, #ffe07b 15%,
                #ffb151 2%, #16104b 50%);
11         }
12     </style>
13 </head>
14 <body>
15 <div class="rainbow-radial-gradient"></div>
16 </body>
17 </html>
```

用浏览器打开 demo3-4，页面效果如图 3-6 所示。

图3-6　demo3-4页面效

在 demo3-4 中，使用径向渐变实现了一个月亮的效果。需要注意的是，圆心坐标的默认值是 center。



多学一招：重复渐变

了解了线性渐变和径向渐变的使用方法后，接下来我们介绍一下重复渐变。重复渐变对以上两种渐变方式是适用，只需在两个属性前添加“repeating-”，具体语法格式如下。

```
/*线性重复渐变*/
repeating-linear-gradient(起始角度, color stop, color stop[, color stop] *)
/*径向重复渐变*/
repeating-radial-gradient(圆心坐标, 渐变形状, 渐变大小, color stop, color stop[, color stop] *)
```

两种重复渐变的效果读者可以自己动手实践。

【项目分析】

有了前导知识作为铺垫，接下来我们分析一下黑马书城页面的结构。黑马书城页面的页面标注和页面结构如图 3-7 和图 3-8 所示。

图3-7　页面标注

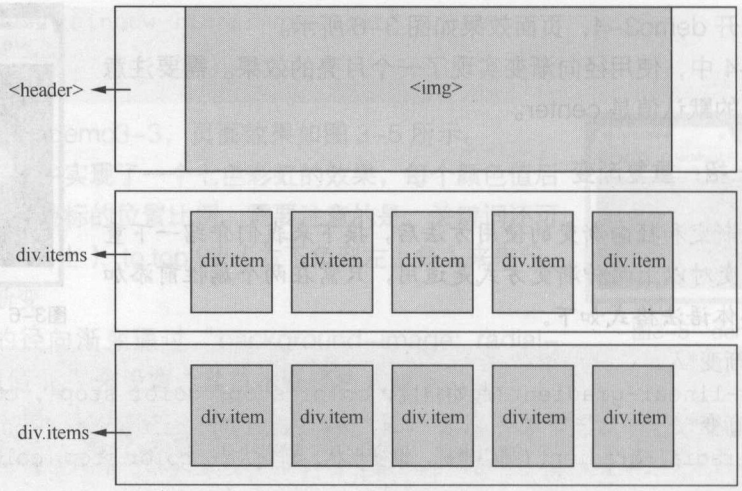


图3-8　页面结构

如图 3-8 所示，黑马书城页面由 header 部分和两个商品模块构成。

该页面的实现细节具体分析如下。

（1）header 部分使用<header>标签中嵌套<img>标签引入图片；两个大的商品模块由类名为 items 的 标签构成，在 div.items 中使用多个类名为 item 的 标签组成多个商品块。

（2）商品信息中 qq 图标为 gif 格式的图片，为了控制不同的文本样式，图书价格信息需要在 中嵌套<i>标签。

（3）在热销教材和精品套系两个模块中使用了渐变色。

（4）为每个图书商品添加阴影，当鼠标悬停在每个商品上面时，显示价格等信息的 层，该层是半透明的效果，使用“background-color: rgba(88,166,240,0.8);”来实现。

（5）rgb 表示 Red Green Bule 3 色，rgba 前 3 个参数表示 3 色的值混合，最后一个参数 0.8 则是指透明度，1 或 100% 表示不透明。

【代码实现】

对项目的结构和样式有所了解后，我们开始编写代码来实现它。HTML 页面代码如 code\03\0301\0301.html 所示。

0301.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>黑马书城</title>
    <link href="css/bookstore.css" type="text/css" rel="stylesheet">
</head>
<body>
  <header><img src="images/tou.png"  alt=""></header>
    <div class="items">
        <div class="item"><div class="p">热销教材</div></div>
        <div class="item">
```

```
13          <div class="pic">
14              <img src="images/1.png" alt="">
15          </div>
16          <div class="desc">
17           <div class="detail"><i>￥36.00</i>[已售7件]
18              <a href="#" ><img border="0" title="联系卖家" alt="联系卖家"
                 src="images/qq.gif"></a>
19           </div>
20          </div>
21      </div>
22      <div class="item">
23          <div class="pic">
24              <img src="images/3.png" alt="">
25          </div>
26          <div class="desc">
27              <div class="detail"><i>￥36.00</i>[已售7件]
28                  <a href="#" ><img border="0" title="联系卖家" alt="联系卖家"
                     src="images/qq.gif"></a>
29              </div>
30            </div>
31      </div>
32      <div class="item">
33          <div class="pic">
34              <img src="images/4.png" alt="">
35          </div>
36          <div class="desc"> <div class="detail"><i>￥36.00</i>[已售7件]
37              <a href="#" ><img border="0" title="联系卖家" alt="联系卖家"
                 src="images/qq.gif"></a>
38          </div>
39            </div>
40      </div>
41      <div class="item">
42          <div class="pic">
43              <img src="images/5.png" alt="">
44          </div>
45          <div class="desc"> <div class="detail"><i>￥36.00</i>[已售7件]
46              <a href="#" ><img border="0" title="联系卖家" alt="联系卖家"
                 src="images/qq.gif"></a>
47          </div>
48          </div>
49      </div>
50  </div>
51   <div class="items">
52      <div class="item"><div class="p">精品套系</div></div>
53      <div class="item">
54          <div class="pic">
55              <img src="images/6.png" alt="">
56          </div>
57          <div class="desc"> <div class="detail"><i>￥36.00</i>[已售7件]
58              <a href="#" ><img border="0" title="联系卖家" alt="" src="images/
```

```
                qq.gif"></a>
59            </div>
60              </div>
61        </div>
62        <div class="item">
63            <div class="pic">
64                <img src="images/7.png" alt="">
65            </div>
66            <div class="desc"> <div class="detail"><i>¥36.00</i>[已售7件]
67                <a href="#" ><img border="0" title="联系卖家" alt="" src=
                  "images/qq.gif"></a>
68            </div>
69            </div>
70        </div>
71        <div class="item">
72            <div class="pic">
73                <img src="images/8.png" alt="">
74            </div>
75            <div class="desc">
76                <div class="detail"><i>¥36.00</i>[已售7件]
77                <a href="#" ><img border="0" title="联系卖家" alt="联系卖家" src=
                  "images/qq.gif"></a>
78                </div>
79            </div>
80        </div>
81        <div class="item">
82            <div class="pic">
83                <img src="images/9.png" alt="">
84            </div>
85            <div class="desc"> <div class="detail"><i>¥36.00</i>[已售7件]
86                <a href="#" ><img border="0" title="联系卖家" alt="联系卖家" src=
                  "images/qq.gif"></a>
87            </div>
88            </div>
89        </div>
90    </div>
91 </body>
92 </html>
```

该项目的 CSS 样式代码如 code\03\css\0301\bookstore.css 所示。

bookstore.css

```
1  /*第3单元  项目3-1 黑马书城*/
2  body {
3      margin: 0;
4      padding: 0;
5      background-color: #F7F7F7;
6  }
7  header,.items{
8      margin-bottom: 20px;
9      width: 1250px;
```

```
    height: 400px;
    padding-left: 20px;
    margin: 10px auto;
}
.items {
    height: 320px;
}
.p {
    background-color: #2A809D;
    height: 320px;
    background-image: linear-gradient(
          to bottom,
          #4b6db9 20%,
          #b2d3ff 80%
    );/*设置该元素背景自上而下地线性渐变*/
    font-family: 'Microsoft Yahei';/*设置字体为微软雅黑*/
    font-size:30px;
    font-weight:bold;
    color:#fff;
    padding-top:40px;
    line-height:30px;/*文本行高为30px*/
}
.item {
    width: 230px;
    height: 300px;
    text-align: center;/*文字水平居中*/
    margin-right: 20px;
    background-color: #FFF;
    float: left;
    position: relative;
    top: 0;
    overflow: hidden;
  /*css3新增动画属性：过渡，all(默认值)指所有属性改变，整个转换过程在0.5s内完成*/
    transition: all .5s;
   /*盒阴影：向下偏移5px，模糊值为5px，颜色为#41a8ff*/
    box-shadow:  #41a8ff 0px 5px 5px ;
}
.pic {
    margin-top: -15px;
    margin-left: -35px;

}
.desc {
    position: absolute;/*绝对定位*/
    bottom: -100px;
    width: 100%;/*宽度是父元素宽度的100%*/
    height: 100px;
    background-color: #58A6F0;
    transition: all .5s;
    background-color: rgba(88,166,240,0.8);
```

```
60
61 }
62 /*当鼠标悬停在该元素上时，该元素绝对定位在父元素顶部-5px 的位置，并且盒阴影为模糊度 15px
   的#AAA 色*/
63 .item:hover {
64     top: -5px;
65     box-shadow: 0 0 15px #AAA;
66 }
67 /*当鼠标悬停在类名为 item 的元素上时，该元素的类名为.desc 的子元素绝对定位，其底部与父元
   素底部对齐*/
68 .item:hover .desc {
69     bottom: 0;
70 }
71 .detail{
72     font-weight: bold;
73     font-size: 20px;
74    margin-top: 50px;
75 }
76 .detail i{color: red;}
```

【项目总结】

本项目的练习重点：

本项目主要练习的知识点有 HTML5 的常用图像标签、CSS 背景设置、CSS 阴影和渐变。

本项目的练习方法：

建议读者在进行 HTML 代码编写时，划清层级，由外到内编写，重复的模块可复制粘贴，再进行细微调整。例如，从结构中可以看出，类名为 items 的 标签有两个，可以先完成一个，再进行复制粘贴。

在进行样式添加时，要着重练习鼠标悬停时的动态效果。

本项目的注意事项：

编写样式代码时回忆前导知识的知识点及每个属性的作用，不要求记住每个属性，但是要对它们有印象，从而在实际开发中灵活使用。

【项目 3-2】 多肉植物商城

【项目描述】

多肉植物由于可以吸收辐射，已成为广受欢迎的室内植物。本项目将带领读者实现一个多肉植物的购物商城页面。与项目 3-1 的黑马书城页面相比，本项目结合 CSS3 的一些新特性，如圆角、过渡、变形等对购物页面进行了一系列的优化，如图 3-9 所示。

【前导知识】

CSS3 的圆角边框

在 CSS3 以前，如果要制作圆角效果，需要在圆角的元素标签中加上 4 个空标签，再在每个

空标签中应用一个圆角的背景，然后对这几个应用了圆角的标签进行相应的定位，这个过程十分麻烦。但是，使用 CSS3 中的 border-radius 属性，实现边框圆角效果就非常简单了。

图3-9 多肉商城

CSS3 的圆角边框实际上是在矩形的 4 个角分别做内切圆，然后通过设置内切圆的半径来控制圆角的弧度，如图 3-10 所示。

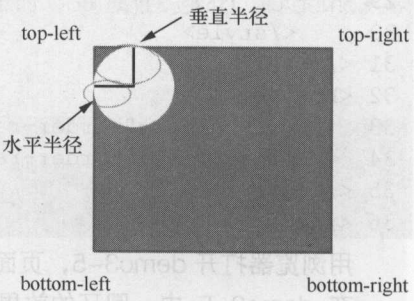


图3-10 矩形的内切圆半径

border-radius 属性的基本语法格式如下。

```
border-radius: 1-4 length|% / 1-4 length|%;
```

在上面的语法中，length 用于设置对象的圆角半径长度，不可为负值。如果“/”前后的值都存在，那么“/”前面的值设置其水平半径，“/”后面的值设置其垂直半径。如果没有“/”，则表示水平和垂直半径相等。另外，其 4 个值是按照 top-left、top-right、bottom-right、bottom-left 的顺序来设置的。如果省略 bottom-left，则其与 top-right 相同；如果省略 bottom-right，则其与 top-left 相同；如果省略 top-right，则其与 top-left 相同。

border-radius 是一种缩写的方式，其实 border-radius 和 border 的属性一样，还可以把各个角单独拆分出来，也就是以下 4 种写法，其参数都是先 y 轴然后 x 轴，具体写法如下。

```
border-top-left-radius: <length>  <length>   //左上角
border-top-right-radius: <length>  <length>  //右上角
border-bottom-right-radius:<length>  <length>  //右下角
border-bottom-left-radius:<length>  <length>   //左下角
```

border-radius 的具体应用如 demo3-5 所示。

demo3-5.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>CSS3 圆角边框</title>
    <style>
        body {
            padding: 0;
            background-color: #F7F7F7;
        }
        div{
            margin:20px;
            float: left;
        }
        /*饼环*/
        .border-radius {
            width: 40px;
            height: 40px;
            border: 70px solid #93baff;
            border-radius: 90px;
        }
        /*四边不同色*/
        .border-radius1 {
            width: 0px;
            height: 0px;
            border-width: 90px;
            border-style: solid;
            border-color: #ff898e #93baff #c89386 #ffb151;
        }
    </style>
</head>
<body>
    <div class="border-radius"></div>
    <div class="border-radius1"></div>
</body>
</html>
```

用浏览器打开 demo3-5，页面效果如图 3-11 所示。

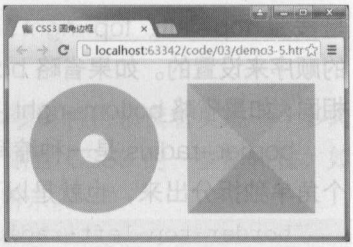


图3-11　demo3-5页面效果

在 demo3-5 中，圆环的效果是将边框的宽度设置得比较宽，border-radius 只有一个值，代表水平和垂直的半径都相等，4 边不同颜色的图形是将矩形的宽高均设置为 0，然后由 4 条边框的不同颜色组成图形，CSS3 圆角边框的属性可以完成很多不同的图形效果，读者可以动手尝试，发挥自己的想象力。

CSS3 的过渡（CSS3 transition）

CSS3 的过渡就是平滑地改变一个元素的 CSS 值，使元素从一个样式逐渐过渡到另一个样式。要实现这样的效果，必须规定两项内容。

（1）规定应用过渡的 CSS 属性名称。

（2）规定效果的时长。

CSS3 的过渡使用 transition 属性来定义，transition 属性的基本语法如下。

```
transition: property duration timing-function delay;
```

transition 是一个复合属性，由 4 个属性构成，如表 3-5 所示。

表 3-5　transition 的子属性

属性	描述	允许取值	取值说明
transition-property	规定应用过渡的 CSS 属性的名称	none	没有属性会获得过渡效果
		all	默认值，所有属性都将获得过渡效果
		property	定义应用过渡效果的 CSS 属性名称列表
transition-duration	定义过渡效果花费的时间	time 值	以秒或毫秒计，默认是 0，意味着没有效果
transition-timing-function	规定过渡效果的时间曲线	linear	规定以相同速度开始至结束的过渡效果（等于 cubic-bezier(0,0,1,1)）
		ease	默认值，规定慢速开始，然后变快，然后慢速结束的过渡效果（cubic-bezier(0.25,0.1,0.25,1)）
		ease-in	规定以慢速开始的过渡效果（等于 cubic-bezier(0.42,0,1,1)）
		ease-out	规定以慢速结束的过渡效果（等于 cubic-bezier(0,0,0.58,1)）
		ease-in-out	规定以慢速开始和结束的过渡效果（等于 cubic-bezier(0.42,0,0.58,1)）
		cubic-bezier(n,n,n,n)	在 cubic-bezier 函数中定义自己的值。可能的值是 0～1 的数值
transition-delay	规定效果开始之前需要等待的时间	time 值	以秒或毫秒计，默认是 0

transition 属性典型的应用就是当用户将鼠标放在一个元素上时 CSS 属性的变化，如 demo3-6 所示。

demo3-6.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>CSS3 过渡</title>
    <style>
        /*方环*/
        .border-radius{
            width: 40px;
            height: 40px;
            border: 70px solid #ff6e9b;
            transition:  3s ;
        }
        /*圆环*/
        .border-radius:hover {
            width: 40px;
            height: 40px;
```

```
            border: 70px solid #ff6e9b;
            border-radius: 90px ;
        }
    </style>
</head>
<body>
    <div class="border-radius"></div>
</body>
</html>
```

用浏览器打开 demo3-6，页面效果如图 3-12 所示。

demo3-6 的页面效果是首先出现一个方环，当鼠标悬停在方环上，方环用 3s 的速度平滑过渡到圆环，这个过程是可以看到的，是不是很像一个动画？没错，transition 属性可以实现简单动画效果的。CSS3 动画的相关属性还有 transform 和 animations，我们在后面的小节中会陆续进行讲解。

图3-12　demo3-6页面效果

CSS3 变形（CSS3 transform）

CSS3 动画相关的第二个属性就是 transform，翻译成中文是“改变，使……变形；转换”，CSS3 transform 属性允许我们对元素进行旋转、缩放、移动或倾斜，向元素应用 2D 或 3D 转换。

transform 属性的基本语法如下。

```
transform: none | transform-functions;
```

在上面的语法中，transform 属性的默认值为 none，适用于内联元素和块元素，表示不进行变形。transform-functions 用于设置变形函数，可以是一个或多个变形函数列表。

1. 2D 转换

transform-functions 常用于 2D 转换的函数说明如表 3-6 所示。

表 3-6　2D 转换的常用函数

函数名	描述	参数说明
rotate(angel)	旋转元素	angel 是度数值，代表旋转角度
skew(x-angel,y-angel)	倾斜元素	angel 是度数值，代表倾斜角度
skewX(angel)	沿着 *x* 轴倾斜元素	
skewY(angel)	沿着 *y* 轴倾斜元素	
scale(x，y)	缩放元素，改变元素的高度和宽度	代表缩放比例，取值包括正数、负数和小数
scaleX(x)	改变元素的宽度	
scaleY(y)	改变元素的高度	
translate(x,y)	移动元素对象，基于 *x* 和 *y* 坐标重新定位元素	元素移动的数值，*x* 代表左右方向，*y* 代表上下方向，向左和向上使用负数，反之使用正数
translateX(x)	沿着 *x* 轴移动元素	
translateY(y)	沿着 *y* 轴移动元素	
matrix(n,n,n,n,n,n)	2D 转换矩阵	使用 6 个值的表示变形，所有变形的本质都是由矩阵完成的

接下来，我们演示一下常用的 2D 转换，如 demo3-7 所示。

demo3-7.html

```
<!DOCTYPE html>
<html>
<head lang="en">
    <meta charset="UTF-8">
    <title>CSS3 变形</title>
    <style>
        .demo{
          margin: 20px;
           padding: 0;
           width: 150px;
           height: 50px;
           background-color: #f3c6ff;
           font-weight: bold;
           font-size: larger;
           float: left;
        }
        .trans1{
           transform:rotate(30deg);
        }
        .trans2{
           transform:skew(30deg);
        }
        .trans3{
           transform:scale(0.8);
        }
        .trans4{
           transform:translate(5px,50px);
        }
    </style>
</head>
<body>
<div class="demo">不设置变形</div>
<div class="demo trans1">rotate(30deg)</div>
<div class="demo trans2">skew(30deg)</div>
<div class="demo trans3">scale(0.8)</div>
<div class="demo trans4">translate(5,50px)</div>
</body>
</html>
```

用浏览器打开 demo3-7，页面效果如图 3-13 所示。

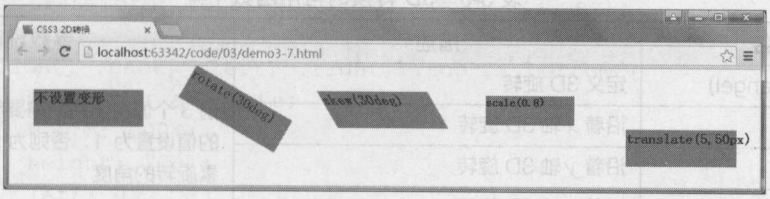


图3-13　demo3-7页面效果

如果一个元素需要设置多种变形效果，可以使用空格把多个变形属性值隔开。

2. 元素的变形原点

元素的变形都有一个原点，元素围绕着这个点进行变形或者旋转，默认的起始位置是元素的中心位置，元素旋转 30° 与旋转前的对比如图 3-14 所示。

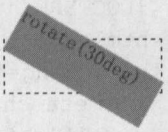


图3-14 元素旋转前后对比

CSS 变形使用 transform-origin 属性指定元素变形基于的原点，语法格式具体如下。

```
transform-origin: x-axis y-axis z-axis;
```

transform-origin 最多接受 3 个值，分别是 *x* 轴、*y* 轴和 *z* 轴的偏移量，transform-origin 的取值说明如表 3-7 所示。

表 3-7 transform-origin 参数

参数	描述	允许取值
x-axis	*x* 轴偏移量	left center right length %
y-axis	*y* 轴偏移量	top center bottom length %
z-axis	*z* 轴偏移量	length

例如，对图 3-14 中旋转 30° 的元素进行如下设置。

```
transform-origin: left 0px 0px;
```

元素旋转前后对比如图 3-15 所示。

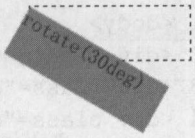


图3-15 元素旋转前后对比

需要注意的是，当参数值为 0px 时，可以省略单位，也可以不写，因为它们都是可选参数。

3. 3D 转换

既然 transform-origin 支持 *z* 轴的偏移，那么 transform 支持 3D 变形也是理所当然的事情。3D 变形是指某个元素围绕其 *x* 轴、*y* 轴、*z* 轴进行旋转。

transform-functions 常用于 3D 转换的函数说明如表 3-8 所示。

表 3-8 3D 转换的常用函数

函数名	描述	参数说明
rotate3d(x,y,z,angel)	定义 3D 旋转	前 3 个值用于判断需要旋转的轴，旋转轴的值设置为 1，否则为 0，angel 代表元素旋转的角度
rotateX(angel)	沿着 *x* 轴 3D 旋转	
rotateY(angel)	沿着 *y* 轴 3D 旋转	
rotate Z(angel)	沿着 *z* 轴 3D 旋转	

续表

函数名	描述	参数说明
scale3d(x,y,z)	定义 3D 缩放	代表缩放比例，取值包括正数、负数和小数
scaleX(x)	沿着 *x* 轴缩放	
scaleY(y)	沿着 *y* 轴缩放	
scaleZ(z)	沿着 *z* 轴缩放	
translate3d(x,y,z)	定义 3D 转化	元素移动的数值
translateX(x)	仅用于 *x* 轴的值	
translateY(y)	仅用于 *y* 轴的值	
translateY(z)	仅用于 *z* 轴的值	
matrix3d(n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n)	3D 转换矩阵	使用 16 个值的 4×4 矩阵。所有变形的本质都是由矩阵完成的
perspective(n)	定义 3D 转换元素的透视视图	一个代表透视深度的数值

由于电脑屏幕是二维平面，表 3-8 中的 perspective 属性就是用于实现视觉上的 3D 效果。接下来，我们看一个典型的案例——立方体，如 demo3-8 所示。

demo3-8.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>CSS 3D转换</title>
    <style>
        body {
            margin: 0;
            padding: 0;
            background-color: #F7F7F7;
        }
        .box {
            width: 200px;
            height: 200px;
            text-align: center;
            line-height: 200px;
            font-size: 24px;
            margin: 100px auto;
            position: relative;
            perspective: 1000px;
            transform-style: preserve-3d;/*定义子元素保留3D位置*/
            transform: rotateX(-30deg) rotateY(30deg);
        }
        .front, .back, .left, .right, .top , .bottom {
            position: absolute;
            width: 100%;
            height: 100%;
            left: 0;
            top: 0;
```

```
        opacity: 0.5;
    }
    .front {
        background-color: pink;
        transform: translateZ(100px);
    }
    .left {
        background-color: green;
        transform: rotateY(90deg) translateZ(-100px);
    }
    .right {
        background-color: red;
        transform: rotateY(-90deg) translateZ(-100px);
    }
    .top {
        background-color: blue;
        transform: rotateX(90deg) translateZ(100px);
    }
    .bottom {
        background-color: yellow;
        transform: rotateX(-90deg) translateZ(100px);
    }
    .back {
        background-color: purple;
        transform: translateZ(-100px);
    }
  </style>
</head>
<body>
<div class="box">
    <div class="front">front</div>
    <div class="back">back</div>
    <div class="left">left</div>
    <div class="right">right</div>
    <div class="top">top</div>
    <div class="bottom">bottom</div>
</div>
</body>
</html>
```

用浏览器打开 demo3-8，页面效果如图 3-16 所示。

在 demo3-8 中，我们给 ".box" 这个 div 元素设置了 perspective 透视，不做透视设置是无法实现立方体效果的；另外，还使用了 transform-style: preserve-3d;属性，用于定义子元素保留 3D 位置，如果不做此项设置，该立方体会是扁的。

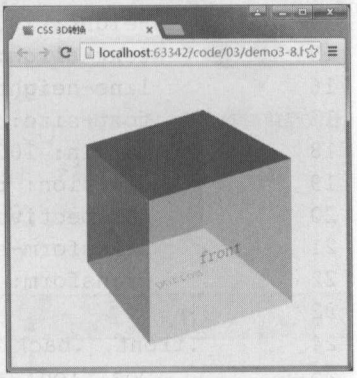


图3-16 demo3-8页面效果

【项目分析】

有了前导知识作为铺垫，接下来我们分析一下多肉植物商城页面的结构。多肉植物商城页面的页面标注和页面结构如图 3-17 和图 3-18 所示。

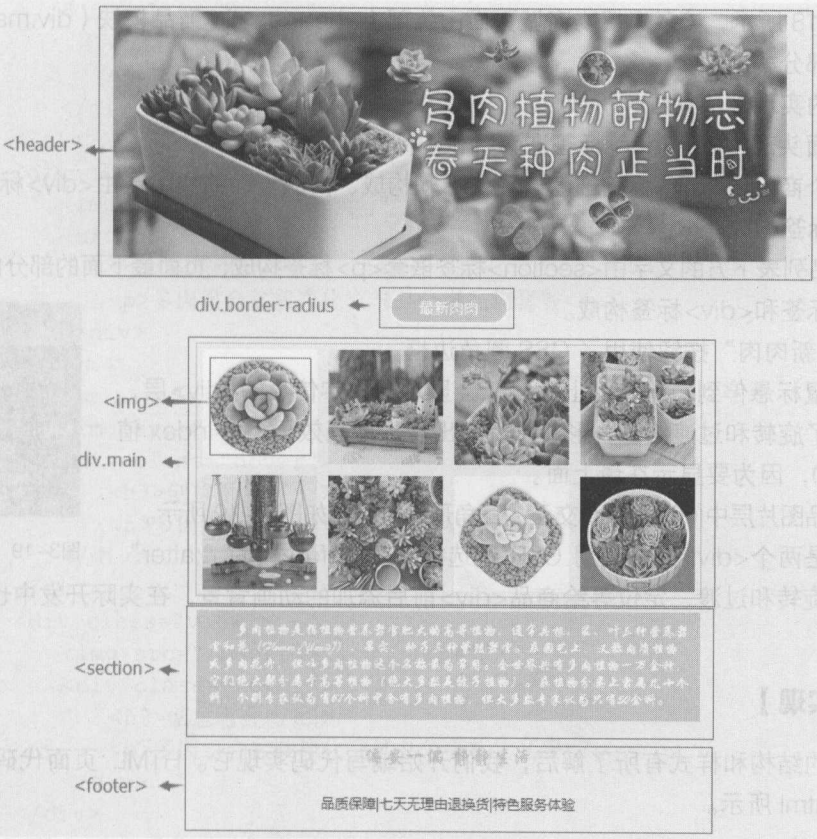


图3-17 页面标注

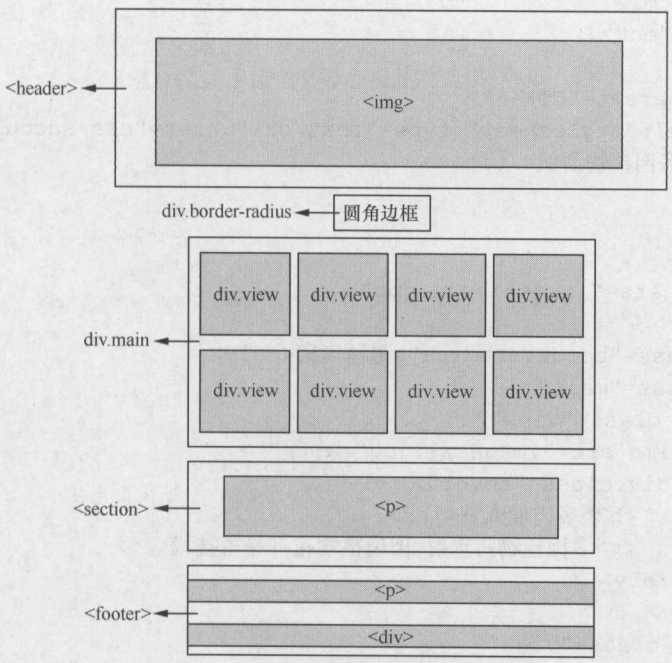


图3-18 页面结构

如图 3-18 所示，该多肉植物商城页面由头部（<header>）、商品模块（div.main）、文字模块和 footer 部分构成。

该页面的实现细节具体分析如下。

（1）页面头部由<header>标签嵌套<img>标签构成。

（2）每个商品由 标签嵌套<img>标签构成、商品上面的文字层由 标签嵌套<h3>标签和<p>标签构成。

（3）商品列表下方的文字由<section>标签嵌套<p>标签构成，页面最下面的部分由<footer>标签嵌套<p>标签和 标签构成。

（4）“最新肉肉”按钮使用了 CSS 圆角边框。

（5）当鼠标悬停到每个商品上面时，会显示出文字信息的 层，该层上应用了旋转和过渡组合，形成了旋转出现的动画效果，z-index 值设置为 4 000，因为要显示在最上面。

（6）商品图片层中间还有两个交汇的三角形的效果，如图 3-19 所示。

图3-19 交汇的三角形

它们不是两个 ，是使用 CSS 伪选择器“:before”和“:after”结合 CSS3 旋转和过渡、定位等给商品 前后添加的动画背景，在实际开发中也是一种常用的方式。

【代码实现】

对项目的结构和样式有所了解后，我们开始编写代码实现它。HTML 页面代码如 code\03\0302\0302.html 所示。

0302.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <link rel="stylesheet" type="text/css" href="css/succulent.css">
    <title>多肉植物商城</title>
</head>
<body>
    <header>
        <img src="images/head.jpg">
    </header>
    <div class="border-radius">最新肉肉</div>
    <div class="main">
        <div class="view">
            <img src="images/1.jpg" alt="">
            <div class="hover">
                <h3>多肉仙人掌</h3>
                <p>多肉植物防辐射 肉肉植物花卉绿植盆栽</p>
            </div>
        </div>
        <div class="view">
            <img src="images/7.jpg" alt="">
```

```
23          <div class="hover">
24              <h3>创意组合</h3>
25              <p>多肉组合盆栽净化空气办公桌礼物套餐</p>
26          </div>
27      </div>
28      <div class="view">
29          <img src="images/3.jpg" alt="">
30          <div class="hover">
31              <h3>创意组合</h3>
32              <p>多肉组合盆栽净化空气办公桌礼物套餐</p>
33          </div>
34      </div>
35      <div class="view">
36          <img src="images/4.jpg" alt="">
37          <div class="hover">
38              <h3>创意组合</h3>
39              <p>多肉组合盆栽净化空气办公桌礼物套餐</p>
40          </div>
41      </div>
42      <div class="view">
43          <img src="images/5.jpg" alt="">
44          <div class="hover">
45              <h3>创意组合</h3>
46              <p>多肉组合盆栽净化空气办公桌礼物套餐</p>
47          </div>
48      </div>
49      <div class="view">
50          <img src="images/6.jpg" alt="">
51          <div class="hover">
52              <h3>创意组合</h3>
53              <p>多肉组合盆栽净化空气办公桌礼物套餐</p>
54          </div>
55      </div>
56          <div class="view">
57          <img src="images/2.jpg" alt="">
58          <div class="hover">
59              <h3>多肉仙人掌</h3>
60              <p>多肉植物防辐射 肉肉植物花卉绿植盆栽</p>
61          </div>
62      </div>
63      <div class="view">
64          <img src="images/8.jpg" alt="">
65          <div class="hover">
66              <h3>创意组合</h3>
67              <p>多肉组合盆栽净化空气办公桌礼物套餐</p>
68          </div>
69      </div>
70  </div>
71  <section>
72      <p>多肉植物是指植物营养器官肥大的高等植物，通常具根、茎、叶 3 种营养器官和花
```

```
(Flower ['flauə]）、果实、种子 3 种繁殖器官；在园艺上，又称肉质植物或多肉花卉，但以多肉植物这个名称最为常用。全世界共有一万余种多肉植物，它们绝大部分属于高等植物（绝大多数是被子植物），在植物分类上隶属几十个科，个别专家认为有 67 个科中含有多肉植物，但大多数专家认为只有 50 余科。
        </p>
    </section>
    <footer>
        <p>偏安一偶  静静生活</p>
        <div class="services">品质保障|七天无理由退换货|特色服务体验 </div>
    </footer>
</body>
</html>
```

该项目的 CSS 样式代码如 code\03\0302\css\ succulent.css 所示。

succulent.css

```
/*第 3 单元  项目 3-2 多肉书城*/
    body {
            margin: 0;
            padding: 0;
            background-color: #F7F7F7;
        }
    header {
        text-align: center;
        height: 450px;
        margin-top: 20px;
    }
    section {
        width: 880px;
        margin: 0 auto;
        background-color: #AAE6DA;
    }
    section>p {
        font-family: 'STXingkai';
        font-size:18px;
        color: #fff;
        line-height:30px;
        padding: 20px;
        text-indent: 2em;/*文本段落首行缩进两格*/
    }
    footer {
        width: 880px;
        margin: 0 auto;
    }
    footer>p {
        font-family: 'STXingkai';
        font-size:35px;
        color: #AAE6DA;
        line-height:20px;
        padding: 20px;
        text-align: center;
    }
```

```
.services {
    font-family: 'Microsoft Yahei';
    font-size:15px;
    color: #374136;
    padding-bottom: 50px;
    text-align: center;
}
    .border-radius {
          width: 200px;
          height: 50px;
          margin: 35px auto;
          background-color: #AAE6DA;
          border:5px solid #fff;
          border-radius: 50px;/*4个边框圆角半径都是50px*/
          font-family: 'Microsoft Yahei';
          font-size:25px;
          color: #fff;
          line-height:50px;
          text-align: center;
    }
    /*当鼠标悬停在该元素上时，绝对定位在父元素顶部-5px，以及给该元素加盒阴影*/
    .border-radius:hover {
          top: -5px;
          box-shadow: 0 0 15px #AAA;/*模糊值为15px的#AAA色的盒阴影*/
       }
    .main {
          width: 880px;
          border: 1px solid #ccc;
          height: 440px;
          margin: 0 auto;
    }
    .view {
          width: 200px;
          height: 200px;
          overflow: hidden;
          position: relative;
          margin: 10px;
          float: left;
       }
       .hover {
          width: 200px;
          background: rgba(0, 0,0,0.5);
          position: absolute;
          top: 40px;
          left: 0;
          text-align: center;
          color: #fff;
          transform: rotate(55deg);/*变形：旋转55度*/
          -webkit-transform: rotate(55deg);
          transition: all 0.5s;/*过渡：所有属性都改变，时长0.5s*/
```

```
87          -webkit-transition: all 0.5s;
88          overflow: hidden;
89          height: 0;
90          z-index: 4000; /*设置元素的堆叠顺序，属性值越大，该元素层离用户越近*/
91      }
92       .hover h3 {
93            color: #fff;
94            border-bottom: 2px solid rgba(76, 179,77, 0.5);
95            padding-bottom: 10px;
96          }
97      .view:hover .hover {
98          height: 120px;
99          transform: rotate(0deg);
100         -webkit-transform: rotate(0deg);
101     }
102     /*before   after 为选择器，一般给选择器加动画背景，设置动画样式*/
103     .view:before {
104         content: "";
105         position: absolute;
106         top: -240px;
107         right: 0;
108         width: 360px;
109         height: 360px;
110         background: rgba( 133, 203,190,0.5);
111         /*变形：旋转 55 度，水平（向右）移动 60px*/
112         transform: rotate(55deg) translateX(60px);
113         -webkit-transform: rotate(55deg) translateX(60px);
114         transform-origin: 100% 0%;/*设置元素变形起始点*/
115         -webkit-transform-origin: 100% 0%;
116         /*过渡动画：所有属性都改变，时长 0.5s，速率逐渐变慢，延迟 0.3s*/
117         transition: all 0.5s ease 0.3s;
118         -webkit-transition: all 0.5s ease 0.3s;
119     }
120     /*写好样式，进行隐藏，用 top:-240px; */
121     .view:hover:before {
122         top: 0;
123     }
124     .view:after {
125         content: "";
126         position: absolute;
127         bottom: -240px;
128         left: 0;
129         width: 360px;
130         height: 360px;
131         background: rgba( 133, 203,190,0.5);
132         transform: rotate(55deg) translateX(-60px);
133         -webkit-transform: rotate(55deg) translateX(-60px);
134         transform-origin: 0% 100%;
135         -webkit-transform-origin: 0% 100%;
136         transition: all 0.5s ease 0.3s;
```

```
137             -webkit-transition: all 0.5s ease 0.3s;
138         }
139         .view:hover:after {
140             bottom: 0px;
141         }
```

【项目总结】

本项目的练习重点：

本项目主要练习的知识点有 CSS3 的圆角边框、过渡和变形。

本项目的练习方法：

建议读者在编码时，要着重练习 hover、before、after 等状态时的过渡和变形效果。对于过渡和变形部分，读者要多练习几遍，有兴趣的读者可以尝试修改其值，从而达到其他的动态效果。

本项目的注意事项：

编写该项目代码时可以与项目 3-1 黑马书城的页面进行对比，因为都是购物页面，会有相似之处，可以帮助读者更好地巩固学习过的知识。

【项目 3-3】 摇晃的桃子

【项目描述】

CSS3 的出现使网页中的动画效果不再只依赖于 flash 和 JavaScript。本项目是一个有趣的动画效果，名称为“摇晃的桃子”，如图 3-20 所示。

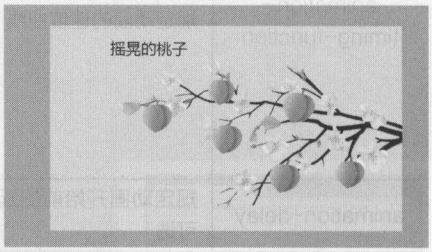


图3-20 摇晃的桃子

【前导知识】

CSS3 动画（CSS3 animations）

在项目 3-2 中，我们学习了 CSS3 的 transition 和 transform 属性，虽然二者结合可以实现一些简单的动画效果，但是也有一些难以克服的缺点，如无法实现动画在多个状态中转换。CSS3 中最后一个动画相关的属性是 animations，有了它就可以解决这样的问题。

一个完整的 CSS animations 由两部分构成：一组定义的动画关键帧和描述该动画的 CSS 声明。接下来，我们分别对其进行说明。

1. @keyframes 规则

在 CSS3 中使用@keyframes 规则来创建动画，keyframes 可以设置多个关键帧，每个关键帧表示动画过程中的一个状态，多个关键帧就可以使动画十分绚丽。

@keyframes 规则的语法格式如下。

```
@keyframes animationname {
        keyframes-selector{css-styles;}
}
```

在上面的语法格式中，animationname 表示当前动画的名称，它将作为引用时的唯一标识，

因此不能为空；keyframes-selector 是关键帧选择器，即指定当前关键帧要应用到整个动画过程中的位置值是一个百分比、from 或者 to。其中，from 和 0%效果相同表示动画的开始，to 和 100%效果相同表示动画的结束；css-styles 定义执行到当前关键帧时对应的动画状态。以上 3 个属性都是必需的，缺一不可。

2. animation 属性

animation 属性用于描述动画的 CSS 声明，包括指定具体动画以及动画时长等行为。

animation 属性的基本语法如下。

```
animation: name duration timing-function delay iteration-count direction
fill-mode play-state;
```

与 transtion 类似，animation 也是一个复合属性，animation 的 8 个子属性如表 3-9 所示。

表 3-9 CSS3 动画 animation 子属性

属性	描述		
animation-name	规定@keyframes 动画的名称	keyframe name	规定需要绑定到选择器的 keyframe 的名称
		none	规定无动画效果（可用于覆盖来自级联的动画）
animation-duration	规定动画完成一个周期所花费的时间	time 值	以秒或毫秒计算，默认是 0
animation-timing-function	规定动画的速度曲线	linear	动画从头到尾的速度是相同的
		ease	默认值。动画以低速开始，然后加快，在结束前变慢
		ease-in	动画以低速开始
		ease-out	动画以低速结束
		ease-in-out	动画以低速开始和结束
		cubic-bezier (n,n,n,n)	在 cubic-bezier 函数中定义自己的值。可能的值是从 0 到 1 的数值
animation-delay	规定动画开始前的延迟，可选	time 值	以秒或毫秒计，默认是 0
animation-iteration-count	规定动画被播放的次数	n	定义动画播放次数的数值，默认是 1
		infinite	规定动画应该无限次播放
animation-direction	规定动画是否在下一周期逆向播放	normal	默认值。动画应该正常播放
		alternate	动画应该轮流反向播放
animation-play-state	规定动画是否正在运行或暂停	paused	规定动画已暂停
		running	默认值，规定动画正在播放
animation-fill-mode	规定对象动画时间之外的状态	none	不改变默认行为
		forwards	当动画完成后，保持最后一个属性值（在最后一个关键帧中定义）
		backwards	在 animation-delay 所指定的一段时间内，在动画显示之前，应用开始属性值（在第一个关键帧中定义）
		both	向前和向后填充模式都被应用

CSS3 动画的具体应用如 demo3-9 所示。

demo3-9.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>CSS3 动画</title>
    <style>
        body {
            margin: 0;
            padding: 0;
            background-color: #F7F7F7;
        }
        .box {
            width: 400px;
            margin: 100px auto;
            animation: rotate 4s linear infinite;
        }
        img {
            width: 100%;
            display: block;
        }
        @keyframes rotate {
            0% {
                transform: rotateZ(0deg);
            }
            100% {
                transform: rotateZ(-360deg);
            }
        }
    </style>
</head>
<body>
<div class="box">
    <img src="./images/fengche.png">
</div>
</body>
</html>
```

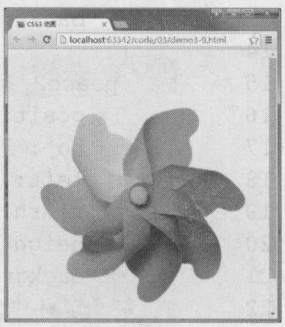


图3-21　demo3-9页面效果

用浏览器打开 demo3-9，页面效果如图 3-21 所示。

在 demo3-9 中，是用@keyframes 规则定义了一个名称为 rotate 的动画，在该动画中定义了两个关键帧，第二个关键帧定义元素围绕 z 轴反转 360°，在第 15 行使用 animation 属性调用了这个动画效果并应用到了包含风车图片的 div 上，形成了风车转动的效果。

CSS 精灵技术（CSS Sprites）

过去，网页开发者喜欢把网页里面的图片字节数控制得非常小，往往在一个图片文件夹里散落着许多小图片，客户端每显示一张图片都会向服务器发送一次请求，图片越多请求次数越多，

这样有可能造成图片延迟加载，影响用户体验。随着互联网技术的发展，大家越来越重视网页的加载速度，于是这些小图片被整合到了一起，CSS Sprites 出现了。

CSS Sprites 被称为 CSS 精灵技术，是一种网页图片应用处理方式。它允许用户将一个页面涉及的所有小图片都包含到一张大图中去，这样一来，当访问该页面时，只需要载入一次图片，然后利用 CSS 的“background-image”“background-repeat”“background-position”组合进行背景定位，将一张大图片中的某个部分显示到网页中的固定位置。

CSS 精灵技术的使用具体如下。

（1）首先要有一张精灵图，它是许多小图片的合并，由于精灵图要求不高于 200KB，所以这种合并适合于一般小图标素材，不适合于大的背景图片，如图 3-22 所示。

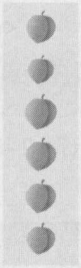


图3-22 精灵图

（2）创建一个容器（如 标签、<i>标签、标签等）来加载精灵图片。

（3）编写 CSS 代码。

在图 3-22 中有 6 个桃子，CSS 精灵技术可以让这些桃子显示在网页中的不同位置，如只在网页中显示前两个桃子，代码如 demo3-10 所示。

demo3-10.html

```
<!DOCTYPE html>
<html>
<head lang="en">
    <meta charset="UTF-8">
    <title>CSS 精灵图</title>
    <style>
        .peach1{
            position: absolute;
            top: 50px;
            left: 0;
            width:90px;
            height:100px;
            background:url(images/demo3-10/peach.png) 0 0 no-repeat;
        }
        .peach2 {
            position: absolute;
            top: 70px;
            left: 100px;
            width : 90px;
            height: 100px;
            background: url(images/demo3-10/peach.png) 0 -110px no-repeat;
        }
    </style>
</head>
<body>
<span class="peach1" ></span>
<span class="peach2" ></span>
</body>
</html>
```

用浏览器打开 demo3-10，页面效果如图 3-23 所示。

在 demo3-10 中，标签作为加载精灵图的容器，使用“position:absolute”对两个标签进行了绝对定位，每个标签的宽度和高度对应精灵图上每个桃子的宽度和高度。需要注意的是，第 13 行和第 21 行应用到了“background-position”设置背景图的位置，背景图在网页中的坐标如图 3-24 所示。

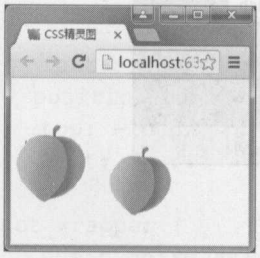


图3-23 demo3-10页面效果

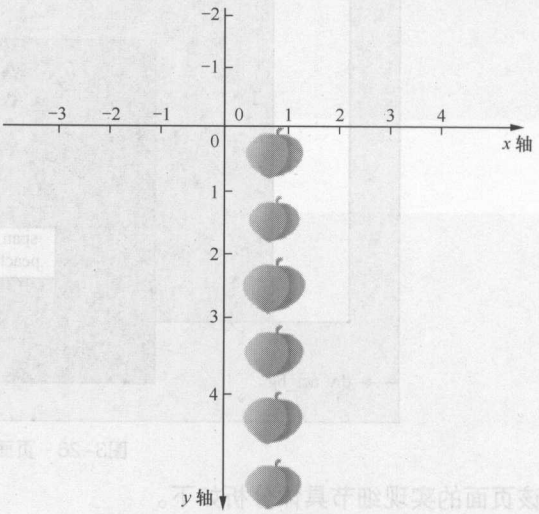


图3-24 背景图在网页中的坐标

从图 3-24 中可以看出，当背景图向上或者向左移动时坐标值取负数，在 demo3-10 中显示第二个桃子需要背景图向上移动，且第一个桃子的高度为 100px，所以第二个桃子的 y 轴取值为“-110px”。

【项目分析】

有了前导知识作为铺垫，接下来我们分析一下摇晃的桃子的页面结构。摇晃的桃子页面的页面标注和页面结构如图 3-25 和图 3-26 所示。

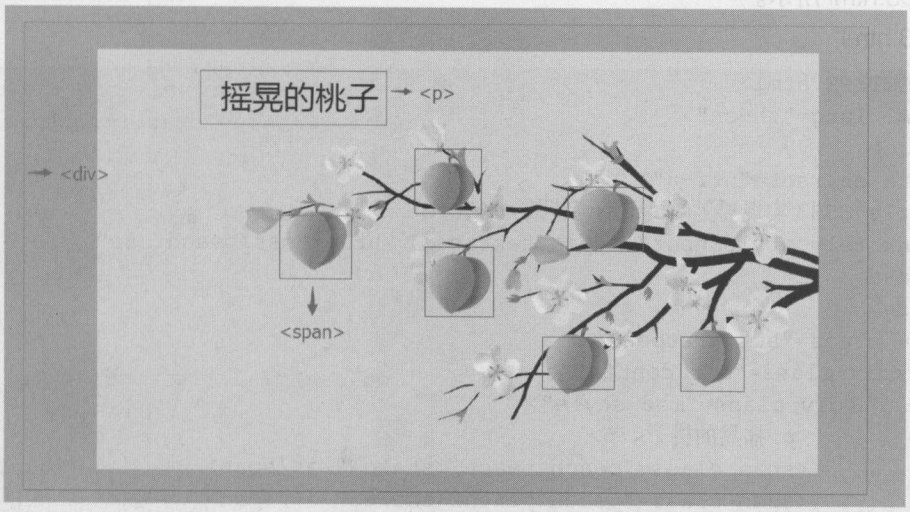


图3-25 页面标注

如图 3-26 所示，该页面通过控制几个 标签的位置，让 标签内的其他标签显示在适当的位置。

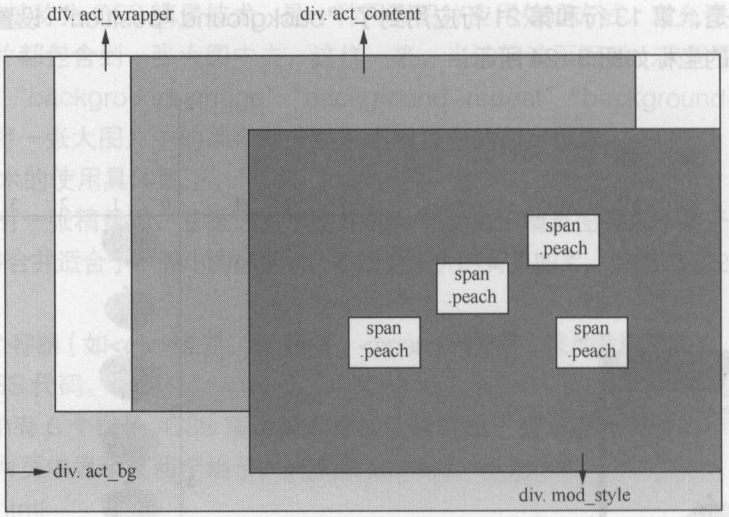


图3-26　页面结构

该页面的实现细节具体分析如下。

（1）为类名为 act_wrapper 的 标签设置背景图，其中包括桃树树枝的部分。

（2）在类名为 act_content 的 标签中嵌套<p>标签，用来添加“摇晃的桃子”文字。

（3）6 个桃子使用标签+CSS3 精灵技术完成，为了对每个桃子定位，需要在标签外层使用 标签。

（4）桃子左右摇晃的效果使用 CSS3 的动画技术来实现。

【代码实现】

对项目的结构和样式有所了解后，即可开始编写代码来实现它。HTML 页面代码如 code\03\0303\0303.html 所示。

0303.html

```
<!DOCTYPE html>
<html lang="zh-cn">
<head>
<meta charset="utf-8">
<title>CSS3 实现摇晃的桃子动画特效</title>
<link rel="stylesheet" type="text/css"  href="css/peach.css">
</head>
<body>
<div class="act_wrapper">
    <div class="act_content">
        <div class="mod_style">
            <p>摇晃的桃子</p>
            <span class="peach peach1 shake1" ></span>
            <span class="peach peach2 shake2" ></span>
            <span class="peach peach3 shake3" ></span>
```

```
            <span class="peach peach4 shake4" ></span>
            <span class="peach peach5 shake5" ></span>
            <span class="peach peach6 shake6" ></span>
        </div>
    </div>
    <div class="act_bg"></div>
</div>
</body>
</html>
```

该项目的 CSS 样式代码如 code\03\0303\css\peach.css 所示。

peach.css

```
/*第 3 单元  项目 3-3 摇晃的桃子*/
p{
    font-family:"微软雅黑";
    font-size: 40px;
   position: absolute;
    top: -100px;
    left: 0px;
}
.act_wrapper {
    position:relative;
    z-index:1;
    min-width:1000px;/*元素最小宽度为 1000px*/
    margin: 0 auto;
    overflow:hidden;
}
.act_wrapper .act_bg {
    position:absolute;
    left:50%;
    top:0;
    z-index:1;
    width:1920px;
    margin-left:-1350px;
}
.act_wrapper .act_content {
    position:relative;
    z-index:2;
    width:1000px;
    height:1200px;
    margin: 0 auto;
    margin-top:-569px
}
.act_bg {
    background:url(../images/bg.jpg) 100% 0 no-repeat;
    height:750px
}
.mod_style {
    position:absolute;
    top:716px;
```

```
    left:200px;
    width:870px;
    height:560px
}
.peach {
    position:absolute;
    top:0;
    left:0;
    width:90px;
    height:100px;
    display:block;
    /*设置背景图片为精灵图片*/
    background:url(../images/peach.png) no-repeat 0 0;
}
.peach1 {
    background-position:0 0;
    top:100px;
    left:72px
}
.peach2 {
    background-position:0 -115px;
    top:39px;
    left:242px
}
.peach3 {
    background-position:0 -215px;
    top:71px;
    left:452px
}
.peach4 {
    background-position:0 -328px;
    top:156px;
    left:261px
}
.peach5 {
    background-position:0 -435px;
    top:256px;
    left:412px
}
.peach6 {
    background-position:0 -545px;
    top:247px;
    left:575px
}

.shake1 {
    -webkit-animation-duration:2.5s;
}
.shake2,.shake6 {
    -webkit-animation-duration:3.5s;
```

```
89 }
90 .shake3 {
91     -webkit-animation-duration:1.5s;
92 }
93 .shake4 {
94     -webkit-animation-duration:4s;
95 }
96 .shake5 {
97     -webkit-animation-duration:3s;
98 }
99 .shake1,.shake2,.shake3,.shake4,.shake5,.shake6 {
100    /*动画被播放的次数：无限次播放*/
101    -webkit-animation-iteration-count:infinite;
102    -webkit-animation-name:shake;/*动画名称：摇晃*/
103    /*动画的速度曲线：以低速开始和结束*/
104    -webkit-animation-timing-function:ease-in-out;
105}
106@-webkit-keyframes shake{
107    0% {
108        -webkit-transform:rotate(2deg);
109        -webkit-transform-origin:50% 0;
110    }
111    20% {
112        -webkit-transform:rotate(10deg);
113        -webkit-transform-origin:50% 0;
114    }
115    40% {
116        -webkit-transform:rotate(0deg);
117        -webkit-transform-origin:50% 0;
118    }
119    60% {
120        -webkit-transform:rotate(-2deg);
121        -webkit-transform-origin:50% 0;
122    }
123    80% {
124        -webkit-transform:rotate(-10deg);
125        -webkit-transform-origin:50% 0;
126    }
127    100% {
128        -webkit-transform:rotate(0deg);
129        -webkit-transform-origin:50% 0;
130    }
131 }
```

【项目总结】

本项目的练习重点：

本项目主要练习的知识点有 CSS3 动画和 CSS 精灵技术。

本项目的练习方法：

建议读者在编码时，先用精灵技术绘制出每个桃子，再给桃子加上动画效果。在加动画效果时，可复制粘贴重复的内容，进行值的修改调整。

本项目的注意事项：

为了得到更真实的动画体验，本项目对每个桃子设置了不一样的动画时间。

【思考题】

1. 请列举 CSS3 中有关动画制作的属性，并说说这些属性给我们带来了哪些好处。
2. 请简述什么是 CSS3 Sprites，如何使用 Sprites。

Responsive Web Design

4 Chapter

单元 4
HTML5 表单的应用

互联网中，网站的核心就是其用户，所以几乎所有网站都会涉及用户登录、注册的页面设计。而表单则作为网页与用户之间最重要的交互工具。之前，表单的使用一直伴随着繁杂的 JavaScript 代码，现在，HTML5 解决了复杂的表单校验。表单也可以说是 HTML5 各种特性中最令人振奋的部分之一。本单元就带领读者使用 HTML5 中的表单制作登录、注册的网页。

【教学导航】

学习目标	1. 掌握 HTML5 表单标签及属性 2. 掌握 HTML5 表单验证
教学方式	本单元主要训练读者对于 HTML5 表单的熟练使用。建议读者先完成前导知识中的表单案例，再完成项目。需要注意的是： 项目 4-1 中的背景可适应各种屏幕； 项目 4-2 中是体验 HTML5 表单的强大验证功能
重点知识	1. HTML5 新增的表单属性 2. <input>标签 3. HTML5 表单验证
关键词	form、autocomplete、novalidate、input、textarea、label、select、datalist、keygen、output、required、pattern

【项目 4-1】 移动版登录页面

【项目描述】

传统行业的商家十分重视门面的装潢，因为一个好的门面会使用户活跃度变得很高。在一个网站中，可以将用户登录视为“门面”，其设计的重要性不言而喻。本项目将带领读者实现一个用户登录页面，如图 4-1 所示。

从图 4-1 可以看出，该界面包含用户名输入框、密码输入框和登录按钮，其中，“用户名”“密码”为提示文字，在用户输入文字后会自动消失，只保留用户输入的文字，获得焦点的文本框颜色会发生变化，如图 4-2 所示。

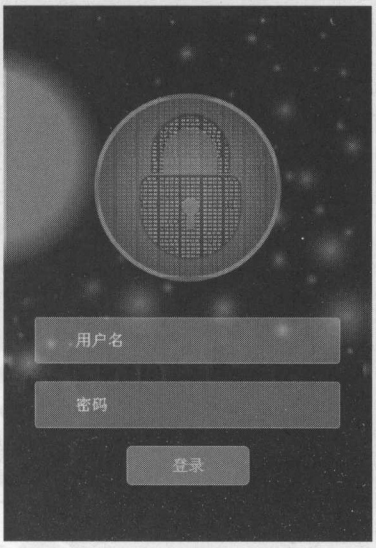


图4-1 用户登录页面

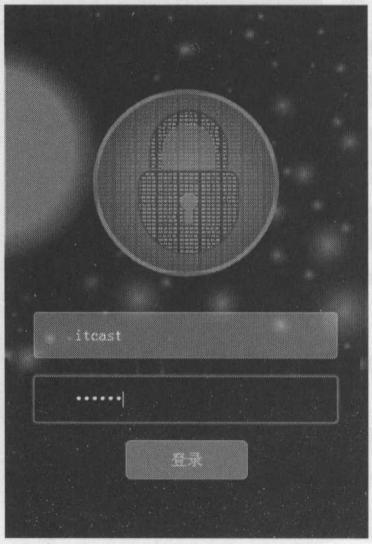


图4-2 用户登录页面

【前导知识】

介绍表单

1. 认识表单

学习表单标签之前，首先需要理解表单的概念。表单主要负责采集用户输入的信息，相当于一个控件集合，由文本域、复选框、单选框、菜单、文件地址域、按钮等表单元素组成。最常见的表单应用有搜索引擎页面、用户登录页面、用户注册页面等。

<form>标签用于创建一个表单，其基本语法格式如下。

```
<form action="url 地址" method="提交方式" name="表单名称">
    各种表单控件
</form>
```

在上面的语法中，action、method、name 为<form>标签的常用属性，name 属性用来区分一个网页中的多个表单；action 属性用于指定接收并处理表单数据的服务器 url 地址；method 属性用于设置表单数据的提交方式，其取值可以为 get 或 post，默认为 get。用这种方式提交的数据将显示在浏览器的地址栏中，保密性差且有数据量限制。而使用 post 不但保密性好，还可以提交大量的数据。明白了两者的差异后，读者可根据实际情况选择使用。

接下来，我们通过创建一个简单的表单来说明表单的结构，如 demo4-1 所示。

demo4-1.html

```
<!DOCTYPE html>
<html>
<head lang="en">
    <meta charset="UTF-8">
    <title>表单</title>
</head>
<body>
<form action="http://www.itcast.cn/" method="post" name="search" >
    <input name="save"/>
    <button>提交按钮</button>
</form>
</body>
</html>
```

用浏览器打开 demo4-1，页面效果如图 4-3 所示。

图4-3 demo4-1页面效果

在 demo4-1 中，虽然代码很少，但是已经包含了表单的 3 个核心元素——表单标签（form）、表单域（input）和表单按钮（button），具体说明如下。

- 表单标签：包含处理表单数据所用的 CGI 程序的 URL 以及将数据提交到服务器的方法。
- 表单域：包含文本框、密码框、隐藏域、多行文本框、复选框、单选框、下拉选择框和文件上传框等。
- 表单按钮：包括提交按钮、复位按钮和一般按钮；用于将数据传送到服务器上的 CGI 脚

本或者取消输入，还可以用表单按钮来控制其他定义了处理脚本的处理工作。

值得一提的是，上述的 CGI 在物理上是一段程序，运行在服务器上，提供同客户端 HTML 页面交互的接口。绝大多数的 CGI 程序被用来解释处理来自表单的输入信息，并在服务器上产生相应的处理，或将相应的信息反馈给浏览器，CGI 程序使网页具有交互功能。

2. HTML5 新增的表单属性

（1）autocomplete 属性

autocomplete 属性用于指定表单是否有自动完成功能。所谓“自动完成”是指将表单控件输入的内容记录下来，当再次输入时，会将输入的历史记录显示在一个下拉列表里，以实现自动完成输入。

autocomplete 属性有 2 个值，具体如下。

- on：表单有自动完成功能。
- off：表单无自动完成功能。

autocomplete 的具体应用如 demo4-2 所示。

demo4-2.html

```
<!DOCTYPE html>
<html>
<head lang="en">
    <meta charset="UTF-8">
    <title>autocomplete 属性</title>
</head>
<body>
<form action="#" method="post" name="search"  autocomplete="true">
    <input name="save"/>
    <button>提交按钮</button>
</form>
</body>
</html>
```

用浏览器打开 demo4-2，首先输入“autocomplete 属性”，单击“提交按钮”，再输入“abc”，单击“提交按钮”，第 3 次输入“a”时的页面效果如图 4-4 所示。

图4-4　demo4-2页面效果

（2）novalidate 属性

novalidate 属性用于指定在提交表单时取消对表单进行有效的检查。为表单设置该属性时，可以关闭整个表单的验证，这样可以使 form 内的所有表单控件不被验证。

HTML5 <input>标签

表单中最为核心的就是<input>标签，使用<input>标签可以在表单中定义文本输入框、单选按钮、复选框、重置按钮等，其基本语法格式如下。

```
<input type="控件类型" />
```

在上面的语法中，type 属性为其最基本的属性，取值有多种，用来指定不同的控件类型。除 type 属性外，还可以定义很多其他属性，常用属性如 name、value、size 等，如表 4-1 所示。

表 4-1　<input>标签相关属性

属性	允许取值	取值说明
type	text	单行文本输入框
	password	密码输入框
	radio	单选按钮
	checkbox	复选框
	button	普通按钮
	submit	提交按钮
	reset	重置按钮
	image	图像形式的提交按钮
	hidden	隐藏域
	file	文件域
	email	E-mail 地址的输入域
	url	URL 地址的输入域
	number	数值的输入域
	range	一定范围内数字值的输入域
	Date pickers (date, month, week, time, datetime, datetime-local)	日期和时间的输入类型
	search	搜索域
	color	颜色输入类型
	tel	电话号码输入类型
name	由用户自定义	控件的名称
value	由用户自定义	input 控件中的默认文本值
size	正整数	input 控件在页面中的显示宽度
readonly	readonly	该控件内容为只读（不能编辑修改）
disabled	disabled	第一次加载页面时禁用该控件（显示为灰色）
checked	checked	定义选择控件默认被选中的项
maxlength	正整数	控件允许输入的最多字符数
autocomplete	on/off	设定是否自动完成表单字段内容
autofocus	autofocus	指定页面加载后是否自动获取焦点
form	form 元素的 id	设定字段隶属于哪一个或多个表单
list	datalist 元素的 id	指定字段的候选数据值列表
multiple	multiple	指定输入框是否可以选择多个值
min、max 和 step	数值	规定输入框所允许的最大值、最小值及间隔
pattern	字符串	验证输入的内容是否与定义的正则表达式匹配
placeholder	字符串	为 input 类型的输入框提供一种提示
required	required	规定输入框填写的内容不能为空

接下来，我们通过一个案例来演示<input>标签的具体使用，如 demo4-3 所示。

demo4-3.html

```
<!DOCTYPE html>
<html>
<head lang="en">
    <meta charset="UTF-8">
    <title>常用表单控件</title>
</head>
<body>
<form action="#" method="post">
    <!--text 单行文本输入框-->
    用户名：<input type="text" value="张三" maxlength="6"><br/><br/>
    <!--password 密码输入框-->
    密码 ：<input type="password" size="40"><br/><br/>
    <!--radio 单选按钮-->
    性别：<input type="radio" name="sex" checked="checked"/>男
    <input type="radio" name="sex"/> 女<br/><br/>
    <!--number 数值输入域-->
    年龄：<input type=number min="18" max="100"><br/><br/>
    <!--checkbox 复选框-->
    兴趣：<input type="checkbox" />唱歌
    <input type="checkbox" />跳舞
    <input type="checkbox" />游泳 <br/><br />
    颜色：<input type="color" value="#ff0000" /><br/><br />
    <!--file 文件域-->
    上传头像：<input type="file" /><br /><br />
    <!--专门用于搜索关键词的文本框-->
    关键词：<input type="search"/><br /><br />
    <!--一定范围内的数值输入域-->
    难易程度：<input type="range" min="1" max="120" /><br/><br/>
    <!--button 普通按钮-->
    <input type="button" value="普通按钮"/>
    <!--submit 提交按钮-->
    <input type="submit" value="提交"/>
    <!--reset 重置按钮-->
    <input type="reset"  value="重置"/>
</form>
</body>
</html>
```

用浏览器打开 demo4-3，使用默认用户名，输入密码，页面效果如图 4-5 所示。

在 demo4-3 中，number、color、search 和 range 为 HTML5 新增的 type 类型。接下来，我们分别介绍一下它们的效果，具体如下。

（1）单击“年龄”输入框测试 number 类型的效果，如图 4-6 所示。

这里，我们设置年龄的范围是 18～100 岁。

（2）单击“颜色：”颜色框，会弹出颜色选取器，如图 4-7 所示。

在颜色选择器中，用户可以自定义自己喜欢的颜色，表单提交时会提交这个颜色的颜色值。

（3）输入关键词“搜索关键词”，搜索框右侧会出现一个“×”按钮，页面效果如图 4-8 所示。单击该按钮，可以清除已经输入的内容。

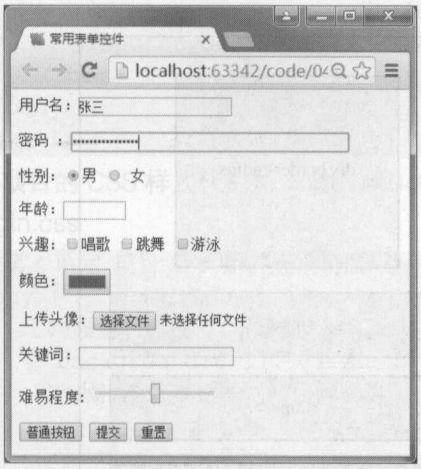


图4-5　demo4-3页面效果

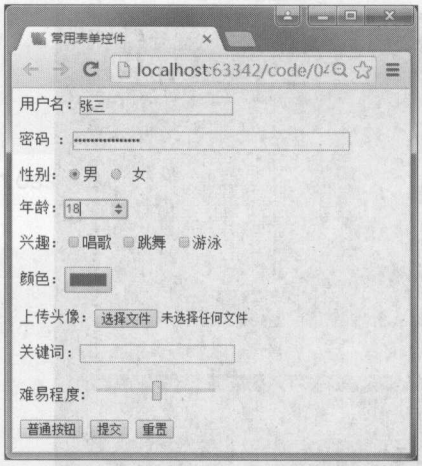


图4-6　number输入框

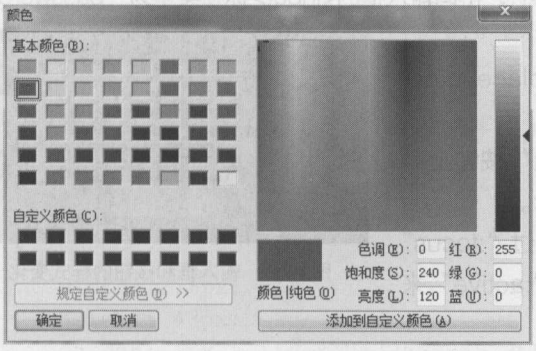


图4-7　颜色选取器

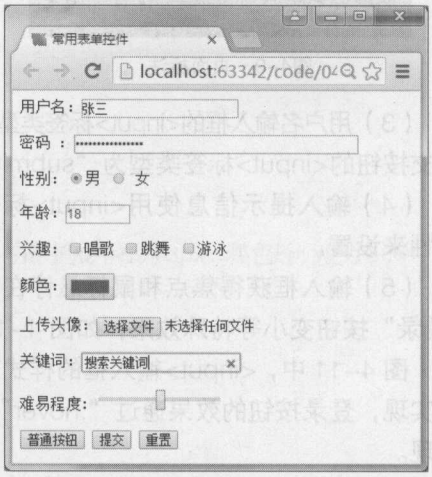


图4-8　关键词输入框

（4）拖动 range 控件可以控制这个数值的大小，这里的数值范围设置为 1¯120。

HTML5 还提供了不同输入类型的文本框，如 email、url。这些文本框会在表单提交时，自动验证输入文本是否符合要求，不符合时会进行错误提示，这部分内容将在项目 4-2 中讲解输入验证时进行详细说明。

【项目分析】

有了前导知识作为铺垫，接下来我们分析一下如何实现用户登录页面。用户登录页面的页面标注和页面结构如图 4-9 和图 4-10 所示。

如图 4-10 所示的用户登录页面由一个<form>标签嵌套表单控件、按钮等部分构成。

该页面的实现细节具体分析如下。

（1）为 body 元素设置背景图片，形成蓝色星空的背景。

（2）使用为 div.border-radius 添加“小锁头”的背景图，并使用 CSS 圆角边框将其变成圆形样式。

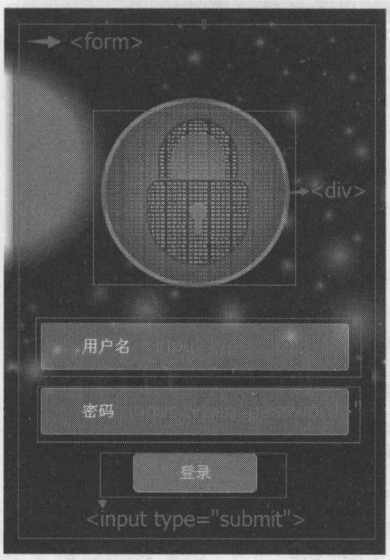


图4-9　页面标注

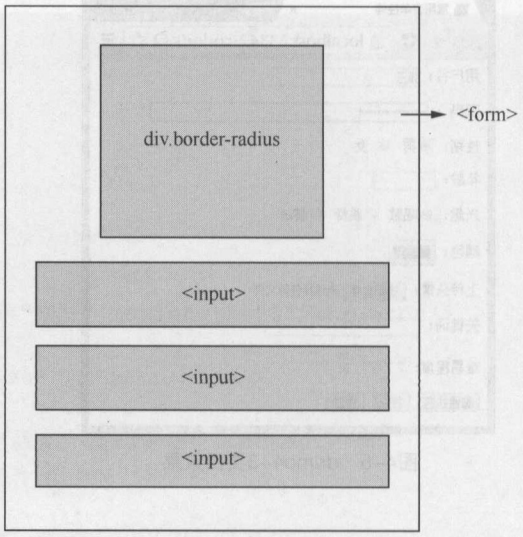


图4-10　页面结构

（3）用户名输入框的<input>标签类型为“text”，密码输入框的<input>标签类型为“password”，提交按钮的<input>标签类型为“submit”。

（4）输入提示信息使用<input>标签的 placeholder 属性来设置。

（5）输入框获得焦点和鼠标悬停在“登录”按钮上时“登录”按钮变小等特殊效果，如图 4-11 所示。

图 4-11 中，<input>输入框的样式变化通过“:focus”来实现，登录按钮的效果通过“:hover” 和“:active”来实现。

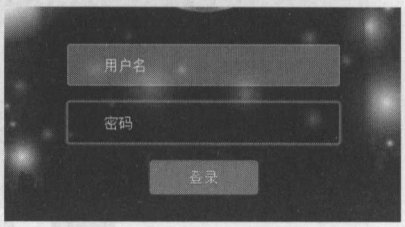


图4-11　输入框和按钮的样式变化

【代码实现】

对项目的结构和样式有所了解后，即可开始编写代码来实现它。该项目的 HTML 页面代码如 code\04\0401\0401.html 所示。

0401.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>用户登录</title>
    <link rel="stylesheet" href="css/login.css" media="screen"
          type= "text/css" />
</head>
<body>
    <form >
        <div class="border-radius"></div>
        <input type="text" name="name" placeholder="用户名" />
```

```
        <input type="password" name="password" placeholder="密码"/>
        <input type="submit" name="submit" value="登录" class="btn" />
    </form>
</body>
</html>
```

该项目的 CSS 样式代码如 code\04\0401\css\login.css 所示。

login.css

```
/*第 4 单元 项目 4-1 用户登录页面*/
body {
      /*设置背景图片 不平铺 水平和垂直都居中 固定不动*/
       background:url(../images/1.png) no-repeat center center fixed;
      /*保持图像本身的宽高比例，将图片缩放到正好完全覆盖定义背景的区域*/
       background-size: cover;
      padding-top:20px;
}
form {
    width: 343px;
    height: 500px;
    margin:0 auto;
    padding:30px;
    border: 1px solid rgba(0,0,0,0.2);
    border-radius: 5px;
    background: rgba(0, 0, 0, 0.5);
/*盒阴影：水平和垂直偏移量都为 0、阴影模糊半径为 13px、阴影扩展半径为 3px、黑色透明度为 50%*/
    box-shadow: 0 0 13px 3px rgba(0,0,0,0.5);
    overflow: hidden;/*隐藏溢出内容*/
}
input {
    width: 276px;
    height: 48px;
    border: 1px solid rgba(255,255,255,0.4);
    border-radius: 4px;
    display:block;
    font-family: 'Source Sans Pro', sans-serif;
    font-size:18px;
    color:#fff;
    padding-left:45px;
    padding-right:20px;
    margin-bottom:20px;
    background: rgba(255, 255, 255, 0.4)  no-repeat 16px 16px;
}
.border-radius {
    width: 200px;
    height: 200px;
    margin: 40px auto;
    background:url(../images/2.jpg) no-repeat center center;
    border:5px solid rgba(255,255,255,0.4);
    border-radius: 200px;
}
```

```
/*设置 当光标放到 type=submit 上时 为一只小手*/
input[type=submit] {
    cursor:pointer;
}
/*设置输入框的提示文字为白色*/
::-webkit-input-placeholder {
      color: #fff;
}
/*当该 input 元素获得焦点时，设置背景颜色及盒阴影*/
input:focus {
      background-color: rgba(0, 0, 0, 0.2);
     box-shadow: 0 0 5px 1px rgba(255,255,255,0.5);
      overflow: hidden;
}
.btn {
    width: 138px;
    height: 44px;
    border-radius: 4px;
   margin:0 auto;
   background: #00B0DC;
   padding: 10.5px 21px;
   border-radius: 6px;
   color: #e1e1e1;
}
/*当鼠标悬停在该元素上时，设置边框、文字阴影、背景颜色及文字颜色*/
.btn:hover {
   border: 1px solid #253737;
   text-shadow: 0 1px 0 #333333;
   background: #00B0DC;
   color: #fff;
}
/*当该元素激活(在鼠标单击与释放之间发生的事件)时，设置外边距、文字阴影、边框、背景颜色及
  文字颜色*/
.btn:active {
   margin-top:1px;
   text-shadow: 0 -1px 0 #333333;
   border: 1px solid #253737;
   background: #00B0DC;
   color: #fff;
}
```

【项目总结】

本项目的练习重点：

本项目主要练习的知识点有 HTML5 的表单和<input>标签。

本项目的练习方法：

本项目的结构非常简单，所以在练习表单标签的同时，读者也要留意本项目中有助于提高用户体验的细节设计，如在适宜的时机加上小手标识、获得焦点的效果变化、当元素被激活时的变化等。

本项目的注意事项：

表单在实际开发中的作用在于数据的提交，所以注意要给表单中用来收集数据的元素取个名字，因为 name 是表单数据提交到服务器后的标识。

【项目 4-2】 用户注册页面

【项目描述】

用户注册是指将用户的相关信息提交到服务器的过程，未注册的用户无权使用网站的一些功能。用户注册页面的设计不需要太过花哨，大方简洁的效果更好。本项目将带领读者完成一个用户注册页面，如图 4-12 所示。

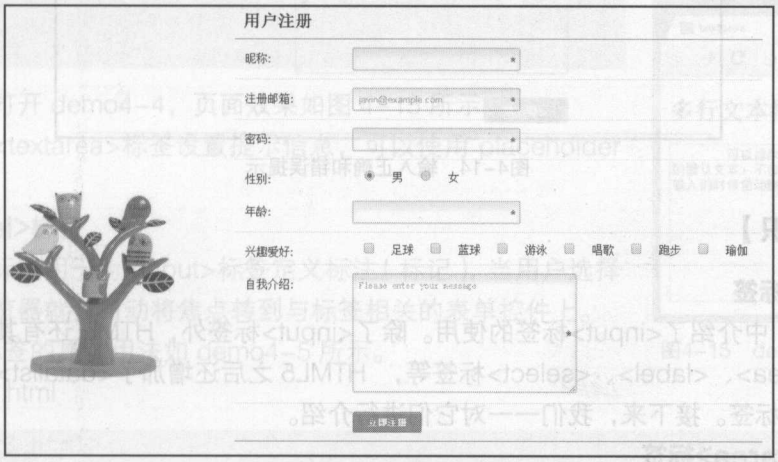


图4-12　用户注册页面

将某个输入框获得焦点后会变宽，未填写正确信息之前会实时显示红色警示图标，如图 4-13 所示。

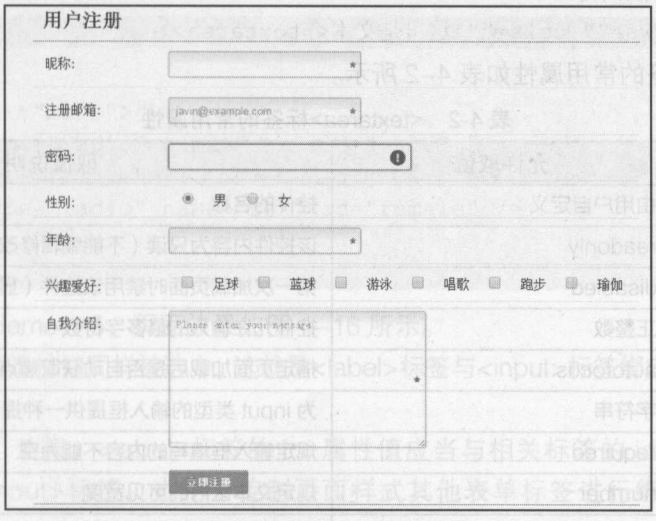


图4-13　输入框获得焦点

填写正确信息和错误提示信息的页面效果如图 4-14 所示。

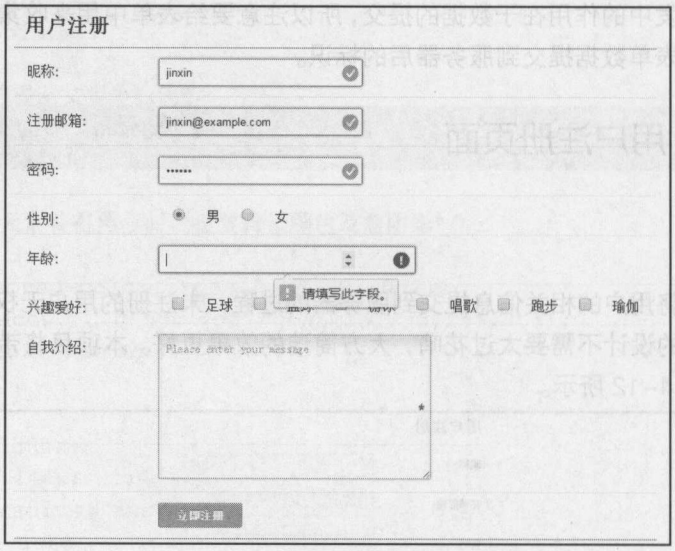


图4-14　输入正确和错误提示

【前导知识】

其他表单标签

项目 4-1 中介绍了<input>标签的使用。除了<input>标签外，HTML 还有其他常用表单标签，如<textarea>、<label>、<select>标签等， HTML5 之后还增加了<datalist>、<keygen>、<output>表单标签。接下来，我们一一对它们进行介绍。

1. <textarea>标签

<textarea>标签用于定义多行文本输入框，可以通过 cols 和 rows 属性来规定文本区域内可见的列数和行数，具体尺寸可以通过 width 和 height 来设置。

其基本语法格式如下。

```
<textarea rows="" cols="">这里是文本</textarea>
```

<textarea>标签的常用属性如表 4-2 所示。

表 4-2　<textarea>标签的常用属性

属性	允许取值	取值说明
name	由用户自定义	控件的名称
readonly	readonly	该控件内容为只读（不能编辑修改）
disabled	disabled	第一次加载页面时禁用该控件（显示为灰色）
maxlength	正整数	控件允许输入的最多字符数
autofocus	autofocus	指定页面加载后是否自动获取焦点
placeholder	字符串	为 input 类型的输入框提供一种提示
required	required	规定输入框填写的内容不能为空
cols	number	规定文本区内的可见宽度
rows	number	规定文本区内的可见行数

<textarea>标签的具体用法如 demo4-4 所示。

demo4-4.html

```
<!DOCTYPE html>
<html>
<head lang="en">
    <meta charset="UTF-8">
    <title>textarea</title>
</head>
<body>
<h2>多行文本框：</h2>
    <textarea name="content"  cols="20" rows="10">
        可以修改或删除的默认文本，不会在用户输入的时候自动删除，
    </textarea>
</body>
</html>
```

用浏览器打开 demo4-4，页面效果如图 4-15 所示。

图4-15 demo4-4页面效果

如果要为<textarea>标签设置提示信息，可以使用 placeholder 属性。

2. <label>标签

<label> 标签用于为<input>标签定义标注（标记），当用户选择该标签时，浏览器就会自动将焦点转到与标签相关的表单控件上。

<label>标签的具体用法如 demo4-5 所示。

demo4-5.html

```
<!DOCTYPE html>
<html>
<head lang="en">
    <meta charset="UTF-8">
    <title>label</title>
</head>
<body>
    性别：
    <label for="male">男</label>
    <input type="radio" name="sex" id="male" />
    <label for="female">女</label>
    <input type="radio" name="sex" id="female" />
</body>
</html>
```

用浏览器打开 demo4-5，页面效果如图 4-16 所示。

单击“女”字，单选按钮同样被选中，这就是<label>标签与<input>标签绑定的作用，如图 4-17 所示。

为达到“绑定”效果，<label>标签的 for 属性值应当与相关标签的 id 属性相同，这里的相关标签不仅指<input>标签，也用于控制页面样式其他表单标签进行绑定，如<textarea>标签等。

图4-16 demo4-5页面效果

图4-17 选中效果

3. <select>标签

<select>标签可创建单选或多选菜单，其语法格式具体如下。

```
<select>
  <option value ="1">选项一</option>
  <option value ="2">选项二</option>
  <option value ="3">选项三</option>
  <option value ="3">选项四</option>
</select>
```

在上面的语法中，<select>标签中的<option>标签用于定义列表中的可用选项。另外，可以通过定义属性，改变下拉菜单的外观显示效果。常用属性如表 4-3 所示。

表 4-3 <select>标签的常用属性

标签名	常用属性	描　述
<select>	size	指定下拉菜单的可见选项数（取值为正整数）
	multiple	定义 multiple="multiple" 时，下拉菜单将具有多项选择的功能，多选方法为按住“Ctrl”键选择多项
<option>	selected	定义 selected ="selected" 时，当前项即为默认选中

<select>标签的具体用法如 demo4-6 所示。

demo4-6.html

```
<!DOCTYPE html>
<html>
<head lang="en">
    <meta charset="UTF-8">
    <title>select</title>
</head>
<body>
<!--单选下拉菜单，可设置默认选中项-->
    所在城市（单选）：<br/>
        <select>
            <option>-请选择-</option>
          <option selected="selected">北京</option>
          <option>上海</option>
          <option>广州</option>
        </select><br/><br/>
    <!--多选下拉菜单，可设置可见选项数，默认选中项可以设置多个-->
    兴趣爱好（多选）:<br />
        <select multiple="multiple" size="4">
          <option>读书</option>
            <option selected="selected">旅行</option>
```

```
            <option selected="selected">听音乐</option>
            <option>运动</option>
        </select>
</body>
</html>
```

图4-18 demo4-6页面效果

用浏览器打开 demo4-6，页面效果如图 4-18 所示。

4. <datalist>标签

<datalist>用于定义输入域的选项列表，通过 id 属性与<input>标签关联，用来配合定义<input>标签可能的值。列表通过<datalist>标签嵌套<option>标签来创建，具体用法如 demo4-7 所示。

demo4-7.html

```
<!DOCTYPE html>
<html>
<head lang="en">
    <meta charset="UTF-8">
    <title>datalist</title>
</head>
<body>
<input id="url" list="urlList">
<datalist id="urlList">
    <option value="www.baidu.com">百度</option>
    <option value="www.sina.com">新浪</option>
    <option value="www.itcast.cn">传智</option>
</datalist>
</body>
</html>
```

用浏览器打开 demo4-7，当鼠标悬停到文本框上时，会出现一个“▼”按钮，如图 4-19 所示。

单击“▼”按钮，会显示列表，如图 4-20 所示。

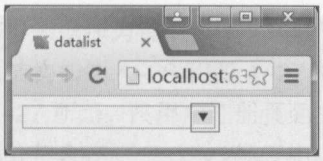


图4-19 demo4-7页面效果

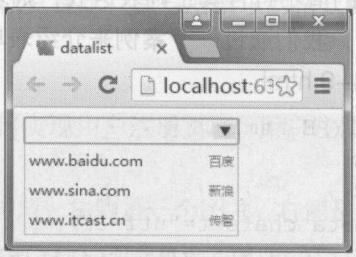


图4-20 demo4-7列表效果

5. <keygen>标签

<keygen>标签是密钥对生成器（key-pair generator）。当提交表单时，会生成两个键，一个是私钥，一个公钥。私钥（private key）存储于客户端，公钥（public key）则被发送到服务器。公钥可用于之后验证用户的客户端证书（client certificate）。

<keygen>标签的具体用法如 demo4-8 所示。

demo4-8.html

```
<!DOCTYPE html>
<html>
<head lang="en">
    <meta charset="UTF-8">
    <title>keygen</title>
</head>
<body>
<form action="#" method="get">
    用户名: <input type="text" name="usr_name" />
    加密强度: <keygen name="security" />
    <input type="submit" />
</form>
</body>
</html>
```

用浏览器打开 demo4-8，单击“加密强度”框右侧的“▼”按钮，页面效果如图 4-21 所示。

图4-21　demo4-8页面效果

6. < output>标签

< output>标签用于定义不同类型的输出，如脚本输出的示例代码如下。

```
<output id="result" onforminput="resCalc()"></output>
```

在上面的语法中，onforminput 属性的值对应脚本的方法名。

HTML5 表单验证

表单验证是一套系统，它为终端用户检测无效的数据并标记这些错误，让 Web 应用更快地抛出错误，大大地优化了用户体验。HTML5 自带的表单验证功能有两种，具体如下。

（1）通过 required 属性校验输入框填写内容不能为空，如果为空将弹出提示框，并阻止表单提交。

（2）通过 pattern 属性规定用于验证 input 域的模式（pattern），它接受一个正则表达式。表单提交时这个正则表达式会被用于验证表单内非空的值，如果控件的值不匹配这个正则表达式就会弹出提示框，并阻止表单提交。那些 type 为 email 或 url 的输入控件内置相关正则表达式，如果 value 的值不符合其正则表达式，则表单将通不过验证，无法提交。

接下来，我们通过一个案例来介绍表单验证的具体情况，如 demo4-9 所示。

demo4-9.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>HTML5 表单验证</title>
</head>
<body>
<form action="#" method="get">
    请输入您的邮箱：<input type="email" name="formmail" required/><br/><br/>
    请输入个人网址：<input type="url" name="user_url" required/><br/><br/>
    <!--pattern 属性用于验证输入的内容是否与定义的正则表达式匹配，正则表达式
     [1-9]d{5}(?!d)代表六位数中国邮编-->
    请输入中国邮编：<input type="text" pattern="[1-9]d{5}(?!d)"
```

```
                    name="postcode" required/><br/><br/>
13      <input type="submit" value="提交"/>
14  </form>
15  </body>
16  </html>
```

用浏览器打开demo4-9，单击“加密强度”框旁边的“提交”按钮，页面效果如图4-22所示。

在图4-22中显示出不能为空的提示，在邮箱输入框中填写“123”，单击“提交”按钮，页面效果如图4-23所示。

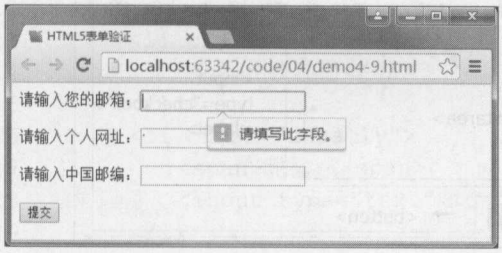


图4-22　demo4-9页面效果

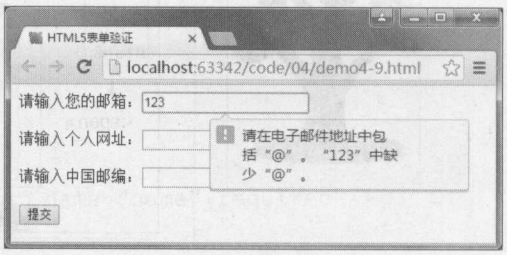


图4-23　错误邮箱提示

填写正确的邮箱信息后，输入错误的个人网址，单击“提交”按钮，页面效果如图4-24所示。

填写正确的网址后，输入错误的邮编，单击“提交”按钮，页面效果如图4-25所示。

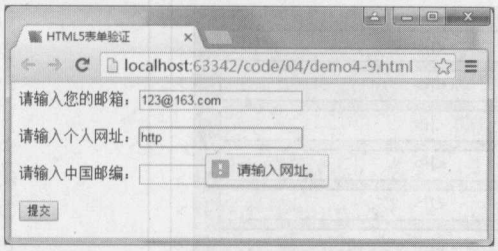


图4-24　错误网址提示

图4-25　正则表达式校验提示

从以上校验过程可以看出，由于邮箱和网址都是HTML5内置的正则校验，所以会进行比较详细的提示，而邮编是我们自定义的正则表达式，所以只提示“请与所请求的格式保持一致”。

【项目分析】

有了前导知识作为铺垫，接下来我们分析一下如何实现用户注册页面。用户注册页面的页面标注和页面结构如图4-26和图4-27所示。

如图4-27所示，该用户注册页面主要由两部分构成，左侧是一个图片，右侧是一个<form>表单，两部分包含在一个 标签中。

该页面的实现细节具体分析如下。

（1）<form>标签中使用<ul>标签会使多个表单控件成列表排列。

（2）使用绝对定位和相对定位控制图片和整个表单在页面上的位置。

（3）每个<li>标签中嵌套表单控件，前面的文字使用标签。

（4）性别和兴趣爱好的选择按钮都使用<label>标签定义标注，单击文字时选择按钮可被选中。

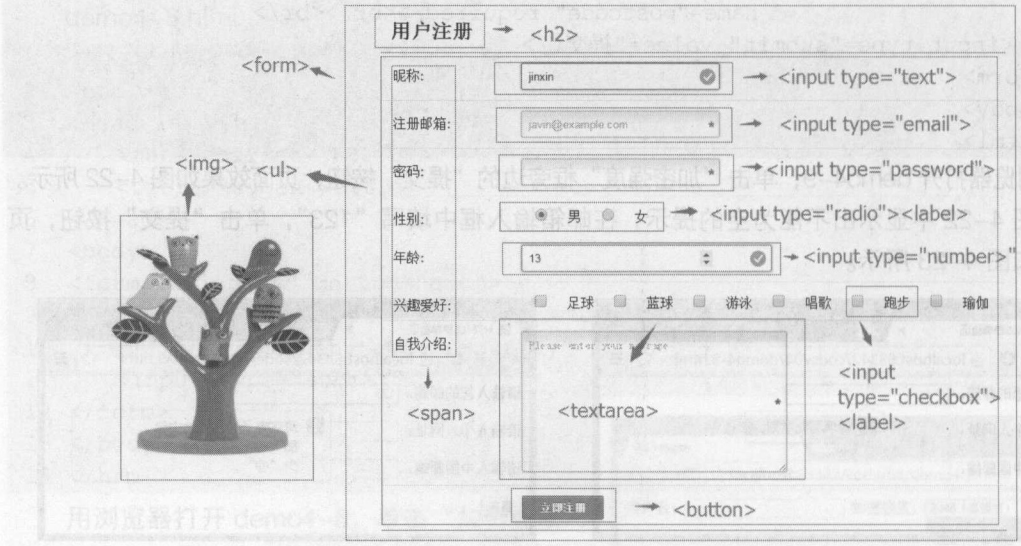


图4-26　页面标注

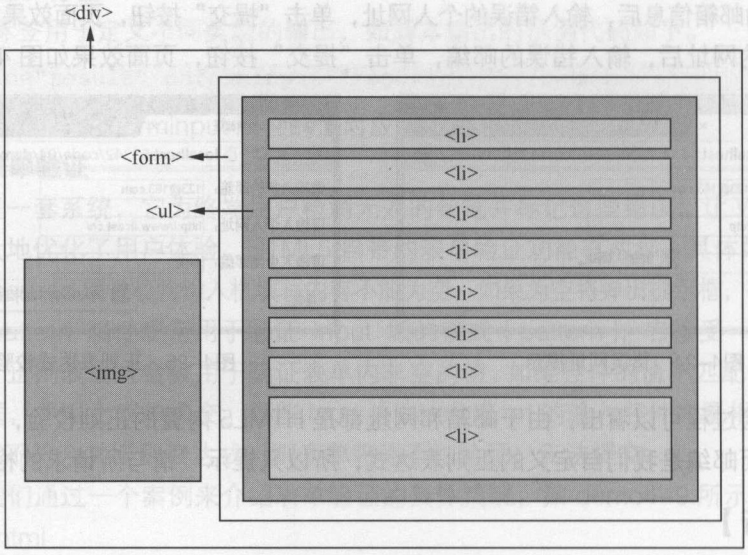


图4-27　页面结构

（5）项目中需要设置页面文字样式、各个表单控件的默认样式、未填写信息的提示样式、填写正确的样式等，其中“ * ”“ ! ”“ ✓ ”都是图片；最后对“提交”按钮进行样式控制。

【代码实现】

对项目的结构和样式有所了解后，即可开始编写代码来实现它。该项目的 HTML 页面代码如 code\04\0402\0402.html 所示。

0402.html

```
1  <!DOCTYPE html>
2  <html>
```

```
<head>
    <meta charset="utf-8">
    <title>用户注册</title>
    <link type="text/css" rel="stylesheet" href="css/register.css">
</head>
<body>
<div>
<img class="tree" src="images/tree.jpg"/>
<form class="contact_form" action="#" method="post" name="contact_form">
    <ul>
        <li class="usually">
            <h2>用户注册</h2>
        </li>
        <li class="usually">
            <span>昵称:</span>
            <input type="text" id="name" name="name" required />
        </li>
        <li class="usually">
            <span>注册邮箱:</span>
            <input type="email"  name="email"
                   placeholder="javin@example. com" required />
        </li>
        <li class="usually">
            <span>密码:</span>
            <input type="password"  name="password" required />
        </li>
        <li class="special">
            <span >性别:</span>
            <input type="radio" name="sex" id="male" checked/>
            <label for="male">男</label>
            <input type="radio" name="sex" id="female" />
            <label for="female">女</label>
        </li>
        <li class="usually">
            <span>年龄:</span>
            <input type="number" name="age" required/>
        </li>
        <li class="special">
            <span >兴趣爱好:</span>
            <input type="checkbox" id="football" name="interest" />
            <label for="football">足球</label>
            <input type="checkbox" id="basketball" name="interest" />
            <label for="basketball">篮球</label>
            <input type="checkbox" id="swim" name="interest" />
            <label for="swim">游泳</label>
            <input type="checkbox" id="sing" name="interest" />
            <label for="sing">唱歌</label>
            <input type="checkbox" id="run" name="interest" />
            <label for="run">跑步</label>
            <input type="checkbox" id="yoga" name="interest" />
```

```
52            <label for="yoga">瑜伽</label>
53        </li>
54        <li class="usually">
55            <span>自我介绍:</span>
56            <textarea rows="10" cols="200" name="introduction"
                        placeholder= "Please enter your message"
57        class="message" required></textarea>
58        </li>
59        <li >
60            <button class="submit" type="submit">立即注册</button>
61        </li>
62    </ul>
63 </form>
64    </div>
65 </body>
66 </html>
```

该项目的 CSS 样式代码如 code\04\0402\css\register.css 所示。

register.css

```
1  /*第 4 单元 项目 4-2 用户注册页面*/
2  *:focus {outline: none;}/*该元素获得焦点时，不出现虚线框（或高亮框）*/
3  .contact_form{
4      width: 70%;/*设置该元素宽度为整个浏览器的 70%*/
5      position: absolute;/*绝对定位*/
6      top: 20%;
7      left: 35%;
8  }
9  .tree{
10     position:relative;/*相对定位*/
11     top: 420px;
12     left: 260px;
13 }
14 .contact_form h2, .contact_form span {
15     font-family:Georgia, Times, "Times New Roman", serif;
16 }
17 .form_hint {font-size: 11px;}
18 .contact_form ul {
19     width:750px;
20     list-style: none;/*清除默认样式*/
21     margin:0px;
22     padding:0px;
23 }
24 .contact_form li{
25     padding:12px;
26     border-bottom:1px solid #eee;
27     position:relative;
28 }
29 /*给类名为 contact_form 的元素的第一个子元素 li 和最后一个子元素 li 加底部边框*/
30 .contact_form li:first-child,
31 .contact_form li:last-child {
```

```
    border-bottom:1px solid #777;
}
.contact_form h2 {
    margin:0;
}
.contact_form span {
    width:150px;
    margin-top: 3px;
    display:inline-block;/*把块元素强制转换为行内块元素*/
    padding:3px;
}
/*给类名为 usually 的元素下的 input 标签定义宽高和内边距*/
.usually input {
    height:20px;
    width:220px;
    padding:5px 8px;
}
.contact_form textarea {padding:8px; width:300px;}
.contact_form button {margin-left:156px;}
.special input {
    height:15px;
    width:40px;
    padding:5px 8px;
}
/*给类名为 usually 的元素下的 input 和 textarea 标签设置背景颜色、背景图片、边框、外阴影、内阴影、边框圆角以及内边距的过渡效果*/
.usually input,.usually textarea {
    background: #fff url(../images/attention.png) no-repeat 98% center;
    border:1px solid #aaa;
    box-shadow: 0px 0px 3px #ccc, 0 10px 15px #eee inset;
    border-radius:2px;
    transition: padding .25s;
}
/*当该元素获得焦点时，设置背景颜色和背景图片、边框、外阴影和右内边距*/
.usually input:focus,.usually textarea:focus {
    background: #fff;
    border:1px solid #555;
    box-shadow: 0 0 3px #aaa;
    padding-right:70px;
}
/* Button Style */
button.submit {
    background-color: #68b12f;
    /*背景颜色线性渐变：由上至下从#68b12f 色过渡到#50911e 色*/
    background: linear-gradient(top, #68b12f, #50911e);
    border: 1px solid #509111;
    border-bottom: 1px solid #5b992b;
    border-radius: 3px;
    box-shadow: inset 0 1px 0 0 #9fd574;/*设置向下 1px 的#9fd574 色的内阴影*/
    color: white;
```

```
    font-weight: bold;/*字体为粗体*/
    padding: 6px 20px;/*内边距：上下 6px、左右 20px*/
    text-align: center;/*文本对齐方式：水平居中*/
    text-shadow: 0 -1px 0 #396715;/*文本阴影向上 1px 的#396715 色*/
}
/*当鼠标悬停在“提交”按钮上时，该按钮背景颜色透明度为 85%，光标变成“小手”形状*/
button.submit:hover {
    opacity:.85;
    cursor: pointer;
}
/*当单击“提交”按钮时，出现 1px 的#20911e 色的实现边框，同时出现内阴影*/
button.submit:active {
    border: 1px solid #20911e;
    box-shadow: 0 0 10px 5px #356b0b inset;
}
/*当该元素获得焦点填写内容无效时，设置背景颜色、背景图片、盒阴影及边框颜色*/
.usually input:focus:invalid,.usually textarea:focus:invalid {
    background: #fff url(../images/warn.png) no-repeat 98% center;
    box-shadow: 0 0 5px #d45252;
    border-color: #b03535
}
/*当该元素获取有效的填写内容时，设置背景颜色、背景图片、盒阴影及边框颜色*/
.usually input:required:valid,.usually textarea:required:valid {
    background: #fff url(../images/right.png) no-repeat 98% center;
    box-shadow: 0 0 5px #5cd053;
    border-color: #28921f;
}
```

【项目总结】

本项目的练习重点：

本项目主要练习的知识点有 HTML 表单标签和 HTML5 的表单验证。

本项目的练习方法：

建议读者完成本项目后动手尝试一下各个表单的验证效果，如邮箱验证，输入“jinxin@163”这样的字符串是可以通过的，但是在实际开发中，有效的邮箱可能需要“jinxin@163.com”这样的格式，应用 HTML5 自带的表单验证要考虑是否能满足需求，如果不能，建议使用正则表达式方式。

【思考题】

1. 请简要介绍表单的 3 个核心元素。
2. 请简述什么 HTML5 的表单验证功能，并列举 HTML5 自带的两种验证方式。

Responsive Web Design

5 Chapter

单元 5
HTML5 画布

在过去的很长一段时间里，网页显示图像是用 jpg、png 等嵌入式图像格式。动画通常是由 Flash 实现的。现在出现了两种新的图形格式 canvas 和 svg，并且 HTML5 对它们提供了非常好的支持，其中，canvas 为 HTML5 的新增元素。本单元将会讲解如何使用 canvas 进行网页图形的绘制。

的 canvas 就是依赖于 JavaScript 才能完成一系列操作。在学习 canvas 之前，我们首先介绍 JavaScript 语言中的基础知识。

1. JavaScript 的引入

在 HTML 文档中引入 JavaScript 与引入 CSS 的方法类似，分为两种方式：一种是在 HTML 文档中直接嵌入 JavaScript 脚本，称为内嵌式；另一种是链接外部 JavaScript 脚本文件，称为外链式，具体如下。

（1）内嵌式

在 HTML 文档中，通过<script>标签及其相关属性可以引入 JavaScript 代码。当浏览器读取到<script>标签时，就执行其中的脚本。其基本语法格式如下。

```
<head>
<script type="text/javascript">
    // 此处为 JavaScript 代码
</script>
</head>
```

在上述语法中，type 属性用来指定 HTML 文档引用的脚本语言类型，当 type 属性的值为“text/javascript”时，表示 script 元素中包含的是 JavaScript 脚本。通常，我们将 script 元素放在<head>和</head>之间，称为头脚本；也可以将其放在<body>和</body>之间，称为体脚本。

（2）外链式

当脚本代码比较复杂或者同一段代码需要被多个网页文件使用时，可以将这些脚本代码放置在一个扩展名为 js 的文件中，然后通过外链式引入该 js 文件。

在 Web 页面中使用外链式引入 JavaScript 文件的基本语法格式如下。

```
<script type="text/javascript" src="JavaScript 文件的路径"></script>
```

2. 数据类型

JavaScript 中有 5 种基本数据类型：Number、String、Boolean、Null 和 Undefined，如表 5-1 所示。

表 5-1　JavaScript 基本数据类型

类型	含义	说　明
Number	数值	正数、负数、0，允许有小数点
String	字符串类型	字符串是用单引号或双引号括起来的一个或多个字符
Boolean	布尔类型	只有 true 或 false 两个值
Null	空类型	没有任何值
Undefined	未定义类型	指变量被创建，但未赋值时所具有的值

除了表 5-1 中的基本数据类型，还有数组（Array）和复杂数据类型 Object，Object 本质上是由一组无序的名值对组成的。

3. 变量

对于每种编程语言来说基础知识都是大同小异的，包括变量、函数、条件语句块、循环语句块等，而对于每种语言的语法却各有不同，如 JavaScript 中在定义变量时，要用 var 进行局部变量的声明，语法格式如下。

```
var str="变量名";
var num=1.5;
    age=23;
var str=new String;
var cars=new Array("A","B","C");//数组
```

“//”为 JavaScript 单行注释，多行注释以“/*”开始、以“*/”结尾。对于弱类型的语言 JavaScript 来说，声明变量可以不加 var，但是，不加 var 该变量将会被识别为全局变量。在 JavaScript 中，“new”关键字用于声明变量，所有的变量均为对象，声明了一个变量时，就创建了一个新的对象。

4. 函数

函数（function）也可以称为方法，是将一些代码组织在一起，形成一个用于完成某个具体功能的代码块，在需要时可以进行重复调用。函数可以大大减少代码的重复，示例代码如下所示。

```
//标准写法
funciton sayHello (){
alert("hello world")
}
//变量形式的写法
var sayHello =function(){
alert("hello world")
}
// 函数可以有参数，它也是弱类型
var sayHello =function(msg){
alert(msg);
}
// 函数的调用
sayHello("hello world");
```

5. 对象

在 JavaScript 中，对象是拥有属性和方法的数据。属性是对象相关的值，方法是对象可以执行的动作。例如，一个叫 lucy 的人在吃饭，我们将这个人视为现实生活中的对象，名字是他的属性，“lucy”为该属性的值，吃饭是该对象的方法，代码如下所示。

```
var person=new Object();//创建对象
    person.name="lucy";//设置 name 属性值为 lucy
    person.eat=function(){
        alert(person.name+"吃饭");
    }
    person.eat();//调用对象方法
```

也可以使用函数的方式来创建对象，代码如下所示。

```
    var person = function(name) {
        this.name = name;
        this.eat=function(){
            alert(name+"吃饭");
        }
    }
    var p=new person("lucy");
    p.eat();
```

6. 事件处理

采用事件驱动是 JavaScript 语言的一个最基本的特征。所谓事件是指用户在访问页面时执行的操作。Event 对象代表事件的状态，如事件在其中发生的元素、键盘按键的状态、鼠标的位置、鼠标按钮的状态。当浏览器探测到一个事件时，如单击鼠标或按键，它可以触发与这个事件相关联的 JavaScript 对象。事件通常与函数结合使用，函数不会在事件发生前被执行。

JavaScript 中常用的事件类型如表 5-2 所示。

表 5-2　JavaScript 中常用事件类型

事件	描　述
onclick	当鼠标单机某个元素时触发此事件
onblur	当前元素失去焦点时触发此事件
onchange	当前元素失去焦点并且元素内容发生改变时触发此事件
onfocus	当某个元素获得焦点时触发此事件
onreset	当表单被重置时触发此事件
onsubmit	当表单被提交时触发此事件
onload	当页面加载完成时触发此事件
onmouseover	鼠标移到某元素之上
onmouseup	鼠标按键被松开
onmousedown	鼠标按钮被按下

接下来，我们通过一个案例来演示在 HTML 中如何调用事件处理程序，如 demo5-1 所示。

demo5-1.html

```
<!DOCTYPE html>
<html>
<head lang="en">
    <meta charset="UTF-8">
    <title>事件处理</title>
    </head>
<body>
     <input type="button" name="btn" value="单击按钮"
    onclick="alert('哎呀！点到我了！');"/>
</body>
</html>
```

用浏览器打开 demo5-1.html，页面效果如图 5-3 所示。

在 demo5-1 中，onclick 代表当前事件类型为鼠标单击事件；alert()函数主要用于弹出警示对话框。单击“单击按钮”，将弹出如图 5-4 所示的警示对话框。

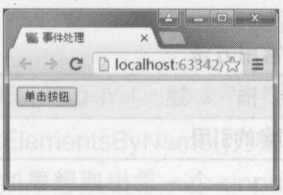


图5-3　demo5-1页面效果

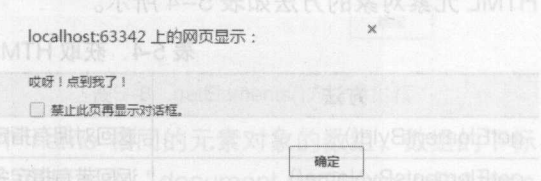


图5-4　警示对话框

另外，事件有很多属性，常用的事件属性如表 5-3 所示。

表 5-3　常用的事件属性

事件	描　述
button	返回当事件被触发时哪个鼠标按钮被单击
clientX	返回当事件被触发时鼠标指针的水平坐标
clientY	返回当事件被触发时鼠标指针的垂直坐标
screenX	返回当某个事件被触发时鼠标指针的水平坐标
screenY	返回当某个事件被触发时鼠标指针的垂直坐标

7. JavaScript HTML DOM

DOM 的全称为文档对象模型（Document Object Model）。当网页被加载时，浏览器会将 HTML DOM 模型构造为对象的树，HTML DOM 树的结构如图 5-5 所示。

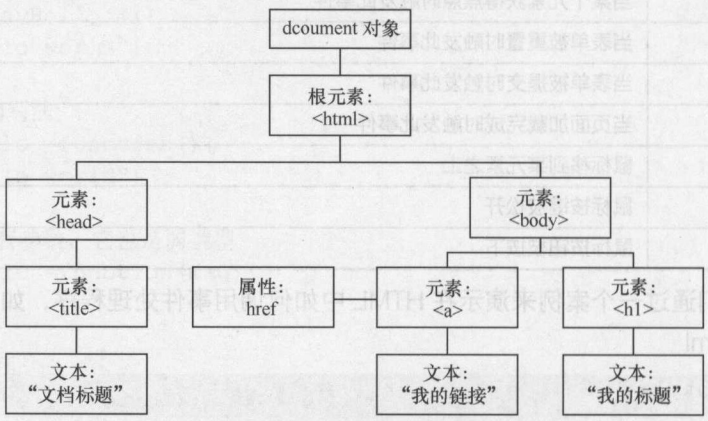


图5-5　DOM对象的文档层次结构图

在图 5-5 中，树的根节点是文档（document）对象，该对象有一个 documentElement 属性，用于表示文档根元素的 Element 对象。HTML 文档中表示文档根元素的 Element 对象是<html>元素，<head>和<body>元素可以被视为树的枝干。通过这个可编程的对象模型，我们就可以通过 JavaScript 来创建动态的 HTML，主要表现在 4 个方面，具体如下。

- JavaScript 能够改变页面中的所有 HTML 元素。
- JavaScript 能够改变页面中的所有 HTML 属性。
- JavaScript 能够改变页面中的所有 CSS 样式。
- JavaScript 能够对页面中的所有事件做出反应。

要想操作 HTML 元素及其属性，首先应该获得这个元素对象，document 对象的常用获取 HTML 元素对象的方法如表 5-4 所示。

表 5-4　获取 HTML 元素对象的常用方法

方法	描　述
getElementById()	返回对拥有指定 id 的第一个对象的引用
getElementsByName()	返回带有指定名称的对象集合
getElementsByTagName()	返回带有指定标签名的对象集合

表 5-4 中的 3 种方法的具体使用如 demo5-2 所示。

demo5-2.html

```
<!DOCTYPE html>
<html>
<head lang="en">
    <meta charset="UTF-8">
    <title>JavaScript 获取 HTML 元素对象</title>
<script type="text/javascript">
    function getValue()
    {
        var x=document.getElementById("input1")
        alert("第一个<input>元素的值为"+x.value);
    }
    function getElements()
    {
        var x=document.getElementsByName("input");
        alert("有"+x.length+"个名称为 input 的元素");
    }
</script>
</head>
<body>
<input  name="input" value="1" id="input1" onclick="getValue()"/><br/>
<input name="input"  value="2"/><br/>
<input name="input"  value="3" /><br/>
<input type="button" onclick="getElements()" value="元素个数"/>
</body>
</html>
```

用浏览器打开 demo5-2.html，页面效果如图 5-6 所示。

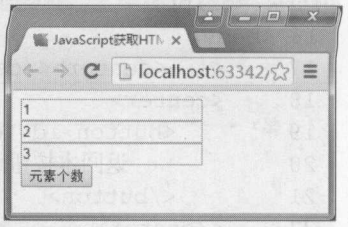


图5-6 demo5-2页面效果

*在 demo5-2 中，3 个<input>标签的名称相同，第一个 id 属*性有值，并且通过 onclick 事件绑定了通过 id 获取值的 getValue()方法；button 按钮绑定了通过 name 获取元素的方法，单击第一个 input 触发 getValue()方法，弹出框如图 5-7 所示。

单击图 5-7 中的“确定”按钮后，单击“元素个数按钮”，触发 getElements()方法，弹出框如图 5-8 所示。

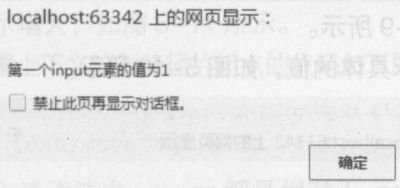


图5-7 getValue()方法弹出框

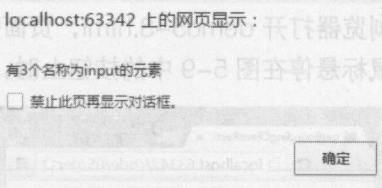


图5-8 getElements()方法弹出框

通过 getElementsByName()方法获取的是一个 name 相同的元素对象的数组，数组的下标从 0 开始。如果想取出第一个<input>标签的值，可以写为“document.getElementsByName("input")[0];”；通常情况下，要获取单个元素的值建议使用“document.getElementsById”。

8. getBoundingClientRect()方法

getBoundingClientRect()方法是 DOM 元素到浏览器可视范围的距离。getBoundingClientRect()最先是 IE 的私有属性，现在已经是一个 W3C 标准。它用于获得页面中某个元素的左、上、右和下分别相对浏览器视窗的位置，或者说一个 Element 元素的坐标。

该方法返回一个 Object 对象，该对象有 6 个属性：top、left、right、bottom、width、height；width、height 是元素自身的宽高，top、left、right、bottom 的大小都是相对于文档视图的左上角来计算的。IE 浏览器只返回 top、left、right、bottom 4 个值，由于本教材推荐使用谷歌浏览器，IE 浏览器的区别这里暂不考虑。

getBoundingClientRect()方法的具体用法如 demo5-3 所示。

demo5-3.html

```
<!DOCTYPE html>
<html>
<head lang="en">
    <meta charset="UTF-8">
    <title>getBoundingClientRect()</title>
</head>
<script>
    /*该方法应用于 button 的 onmousemove 事件*/
    function getRect(){
       var obj= document.getElementById("example");//获取元素对象
        var objRect=obj.getBoundingClientRect();//获取按钮位置
        //当调用该方法时弹出信息
        alert("top:" + objRect.top + ", right:" + objRect.right + ", bottom:"
                    + objRect.bottom + ", left:" + objRect.left);
    }
</script>
<body>
    <!--<center></center>表示将标签内所有的内容居中-->
    <center>
        <button id="example" onmousemove="getRect()" >
            返回本按钮距离浏览器左上角的值
        </button>
    </center>
</body>
</html>
```

用浏览器打开 demo5-3.html，页面效果如图 5-9 所示。

当鼠标悬停在图 5-9 中的按钮上时，弹出框显示具体的值，如图 5-10 所示。

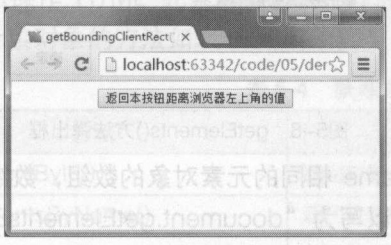


图5-9　demo5-3页面效果

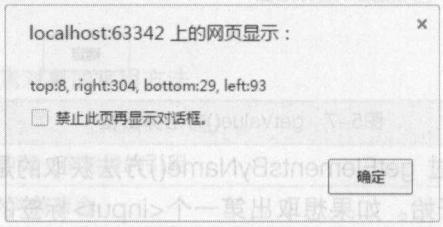


图5-10　demo5-3弹出框

初识 canvas

canvas 意为画布，现实生活中的画布是用来作画的，HTML5 中的 canvas 与之类似，我们可以称它为“网页中的画布”，有了这个画布便可以轻松地在网页中绘制图形、文字、图片等。

1. 创建画布

HTML5 中提供了<canvas>标签，使用<canvas>标签可以在网页中创建一个矩形区域的画布，它本身不具有绘制功能，可以通过脚本语言（如 JavaScript）操作绘制图形的 API 进行绘制操作。

在网页中创建一个画布的语法如下所示。

```
<canvas id="cavsElem" width="400" height="300">您的浏览器不支持 canvas</canvas>
```

在上面的语法中，定义 id 属性是为了在 JavaScript 代码中引用元素。标签中间的文字在浏览器不支持 canvas 的情况下才会显示。<canvas>标签与<img>标签一样，有两个原生属性 width 和 height，默认 300 × 150px，没有单位的值将会被忽略不计。另外，不要用 CSS 控制它的宽和高，否则可能会导致画布上的图形变形。

要在画布中绘制图形，首先要通过 JavaScript 的 getElementById()函数获取网页中的画布对象，代码如下所示。

```
var canvas = document.getElementById('cavsElem');
```

canvas 画布默认为透明，背景色可以自定义。

2. 准备画笔

有了画布之后，要开始作画需要准备一只画笔，这只画笔就是 context 对象，context 对象是画布的上下文，也称为绘制环境，是所有的绘制操作 API 的入口。该对象可以使用 JavaScript 脚本获得，具体语法如下所示。

```
var context = canvas.getContext('2d');
```

在上面的语法中参数“2d”代表画笔的种类，这里用来执行二维操作。三维操作也是存在的，可以把参数替换为“webgl”。三维操作目前还没有被广泛地应用，作为了解即可。

3. 坐标和起始点

2d 代表一个平面，绘制图形时需要在平面上确定起始点，也就是“从哪里开始画”，这个点需要通过坐标来控制，canvas 的坐标系从最左上角“0，0”开始。*x* 轴向右表示增大，*y* 轴向下表示增大，如图 5-11 所示。

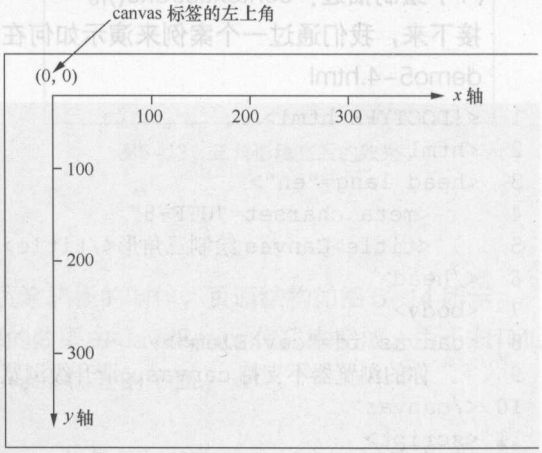


图5-11 canvas的坐标系说明图

设置上下文绘制路径的起点的代码如下所示：

```
var context = canvas.getContext('2d');
context.moveTo(x,y);
```

在上述语法中，x、y 都是相对于 canvas 画布的最左上角。使用 context 对象的 moveTo()方法进行设置，相当于移动画笔到某个位置。

4. 绘制线条

在 canvas 中使用 lineTo()方法绘制直线，代码如下所示。

```
context.lineTo(x,y);
```

在上面的语法中，“x,y”为线头点坐标，lineTo()方法用于定义从“x,y”的位置绘制一条直线到起点或者上一个线头点。

5. 路径

路径是所有图形绘制的基础。例如，绘制直线确定了起始点和线头点后，便形成了一条绘制路径。如果是复杂路径的绘制，必须使用开始路径和闭合路径方法，代码如下所示。

```
context.beginPath(); /*开始路径*/
context.closePath(); /*闭合路径*/
```

开始路径的核心作用是将不同线条绘制的形状进行隔离，每次执行此方法，表示重新绘制一个路径，跟之前的绘制路径可以进行分开样式的设置和管理；闭合路径会自动把最后的线头和开始的线头连在一起。

6. 描边

在 canvas 图形绘制中，路径只是草稿，真正的绘制线必须执行 stroke()方法根据路径进行描边，代码如下所示。

```
context.stroke();
```

有了以上内容作为基础，我们就可以利用 canvas 来绘制一个图形了，基本步骤总结如下。

（1）创建画布：<canvas></canvas>。

（2）准备画笔（获取上下文对象）：canvas.getContext('2d');。

（3）开始路径规划：context.beginPath();。

（4）移动起始点：context.moveTo(x, y);。

（5）绘制线（矩形、圆形、图片等）：context.lineTo(x, y);。

（6）闭合路径：context.closePath();。

（7）绘制描边：context.stroke();。

接下来，我们通过一个案例来演示如何在页面绘制一个三角形，如 demo5-4 所示。

demo5-4.html

```
<!DOCTYPE html>
<html>
<head lang="en">
    <meta charset="UTF-8">
    <title>Canvas 绘制三角形</title>
</head>
<body>
<canvas id="cavsElem">
    你的浏览器不支持 canvas，请升级浏览器
</canvas>
<script>
    //===============基本绘制 API====================
    //获得画布
    var canvas = document.getElementById('cavsElem');
    var context = canvas.getContext('2d');  //获得上下文
    //设置标签的属性宽高和边框
    canvas.width = 900;
```

```
    canvas.height = 600;
    canvas.style.border="1px solid #000";
    //绘制三角形
    context.beginPath();        //开始路径
    context.moveTo(100,100);    //三角形，左顶点
    context.lineTo(300, 100);   //右顶点
    context.lineTo(300, 300);   //底部的点
    context.closePath();        //结束路径
    context.stroke();           //描边路径
</script>
</body>
</html>
```

用浏览器打开 demo5-4，页面效果如图 5-12 所示。

在 demo5-4 中，使用 JavaScript 为画布设置了宽高和边框，然后通过坐标确定了三角形的3个点，规划了绘制路径，最后进行描边操作，成功地绘制了一个三角形。

7. 填充

在 demo5-4 中绘制了一个只有边框的空三角形，canvas 中提供了用于填充当前图形（闭合路径）的方法，具体代码如下所示。

```
context.fill();
```

在 demo5-4 中进行描边操作之后添加上述填充方法，页面效果如图 5-13 所示。

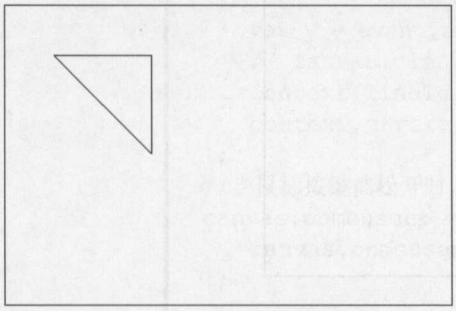


图5-12 demo5-4页面效果

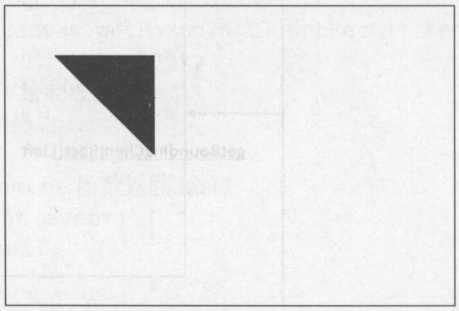


图5-13 三角形填充后的效果

【项目分析】

有了前导知识作为铺垫，接下来我们分析网页涂鸦板的制作，页面结构如图 5-14 所示。

该页面主要应用 HTML<canvas>标签，涂鸦的效果由 JavaScript 代码来完成，由于 HTML 是逐行加载的，所以 JavaScript 代码要写在<canvas>标签的下面。

该页的实现细节具体分析如下。

（1）该涂鸦板要在屏幕中间显示，所以<canvas>标签可以嵌套在 标签中。

（2）编写 JavaScript 代码，首先要创建画布、准备画笔，并且为画布设置宽高和边框；然后将鼠标指针看成画笔，当鼠标按下触发 onmousedown 事件时使用 moveTo()方法确定起点，当鼠标移动触发 onmousemove 事件时使用 lineTo()进行划线。

（3）获取鼠标的 X、Y 坐标很简单，可以使用 clientX 和 clientY 来获取，如图 5-15 所示。

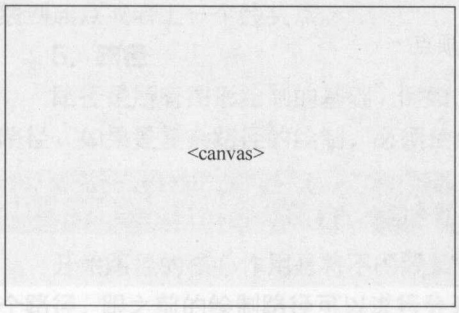


图5-14　页面结构

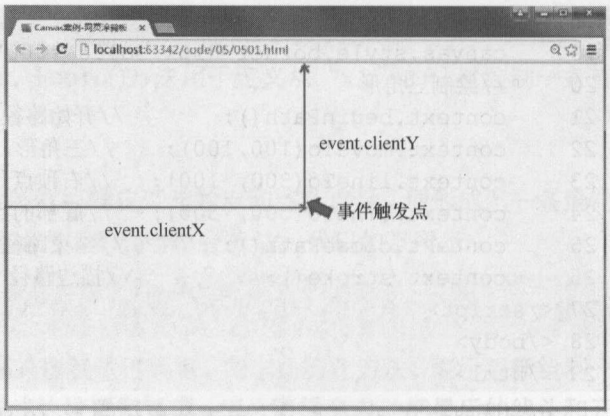


图5-15　获取鼠标坐标

图 5-15 所示的是鼠标直接作用于浏览器窗口的情况，但是当鼠标作用在一个对象（如画布）上时，就要考虑这个对象在浏览器窗口中的位置，这时便要使用 getBoundingClientRect()方法来获取 canvas 矩形对象，并且使用鼠标的坐标减去这个矩形对象到浏览器左上角的距离，如图 5-16 所示。

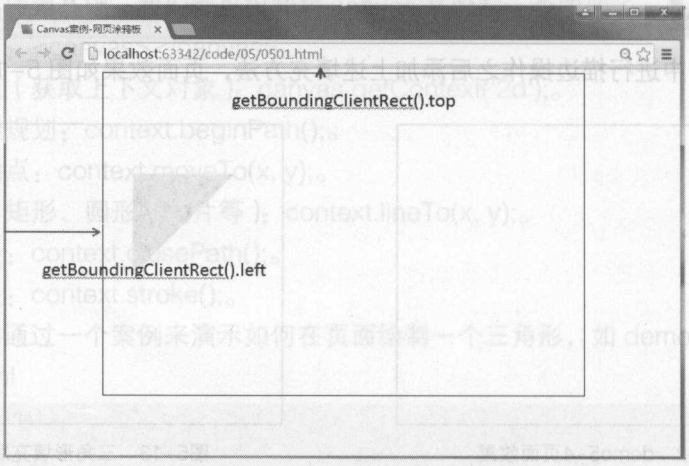


图5-16　画布到浏览器左上角的距离

【代码实现】

接下来，即可开始编写代码来实现该项目。该项目的页面代码如 code\05\0501\0501.html 所示。

0501.html

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>Canvas 案例-网页涂鸦板</title>
    </head>
<body>
    <!--<center></center>表示将标签内所有的内容居中-->
```

```
    <center>
      <canvas id="cavsElem">
       你的浏览器不支持 canvas，请升级浏览器
      </canvas>
    </center>
    <script>
       (function(){
          //获得画布
          var canvas = document.getElementById('cavsElem');
          //准备画笔（获取上下文对象）
          var context = canvas.getContext('2d');
          //设置标签的宽高和边框
          canvas.width = 900;
          canvas.height = 600;
          canvas.style.border = "1px solid #000";
          //当鼠标按下触发 onmousedown 事件时，定义一个函数获取鼠标起始坐标
          canvas.onmousedown = function(e) {
             var x = e.clientX - canvas.getBoundingClientRect().left;
             var y = e.clientY - canvas.getBoundingClientRect().top;
             context.beginPath();//开始规划路径
             context.moveTo(x, y);//移动起始点
             //当鼠标移动触发 onmousemove 事件时，定义一个函数获取绘制线条的坐标
             canvas.onmousemove = function(event) {
                var x = event.clientX-
                        canvas.getBoundingClientRect().left;
                var y = event.clientY- canvas.getBoundingClientRect().top;
                // canvas.clearRect(0, 0, 900, 600);
                context.lineTo(x, y);//绘制线条
                context.stroke();//绘制描边
             };
             //当鼠标按键被松开时，onmousemove 函数返回 null
             canvas.onmouseup = function(event) {
                canvas.onmousemove = null;
             };
          };
       }());
    </script>
</body>
</html>
```

【项目总结】

本项目的练习重点：

本项目主要练习的知识点是回顾 JavaScript 基础和 canvas 绘制简单图形的步骤。

本项目的练习方法：

建议读者在编码前，要将 canvas 绘制的步骤牢记于心：画布、画笔、开始路径、移动坐标、绘制。以该步骤作为思路，完成本项目的编码。

本项目没有添加颜色和画笔粗细等功能是为了让读者更清晰地掌握 canvas 绘制图形的步骤，避免代码复杂而产生误区。在清楚了本项目的实现后，读者可以尝试添加一些功能。

【项目 5-2】 发红包才能看的照片

【项目描述】

如今，用手机发红包已经是家喻户晓的事，社交软件中经常出现好友之间“要红包”的情况，甚至出现了“发红包才能看的照片”，一些好奇心强的小伙伴为了看照片就必须发红包，本项目将带领读者揭开它的真面目。

红包照片的初始状态如图 5-17 所示。

在图 5-17 中，可以看到一张模糊不清的照片，只有一个圆形区域可以看清，单击“想看我么”按钮后，图中的圆形区域会移动到其他位置，如图 5-18 所示。

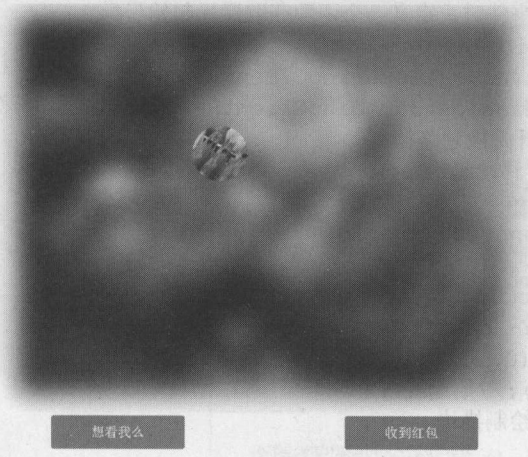


图5-17 发红包才能看的照片

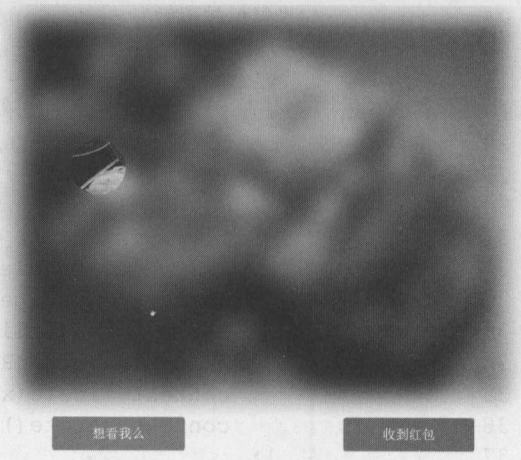


图5-18 移动的圆形区域

圆圈的移动可以勾起用户的好奇心，单击“收到红包”按钮，就可以看到照片了，如图 5-19 所示。

图5-19 收到红包

【前导知识】

canvas 绘制矩形和清除矩形

在项目 5-1 中，介绍了 canvas 绘制图形的基本步骤，getContext("2d") 对象作为 HTML5 的内建对象，还提供了快速绘制矩形、圆形、字符以及添加图像的方法。例如，分别使用 strokeRect()方法和 fillRect()方法来绘制矩形边框和填充矩形，代码如下所示。

```
context.strokeRect(x,y,width,height);//绘制矩形边框
context.fillRect(x,y,width,height); //绘制填充矩形
```

在上面的语法中，*x*、*y*代表矩形起点的横纵坐标，width 和 height 代表要绘制矩形的宽和高，需要注意的是两个方法是可以单独使用的，如 demo5-5 所示。

demo5-5.html

```
<!DOCTYPE html>
<html>
<head lang="en">
    <meta charset="UTF-8">
    <title>绘制矩形</title>
</head>
<body>
<canvas id="cavsElem">
    你的浏览器不支持 canvas，请升级浏览器
</canvas>
<script>
    //===============绘制矩形====================
    //获得画布
    var canvas = document.getElementById('cavsElem');
    var context = canvas.getContext('2d');  //获得上下文
    //设置标签的属性宽高和边框
    canvas.width = 900;
    canvas.height = 600;
    canvas.style.border="1px solid #000";
    //绘制矩形
    context.strokeRect(0,0,200,100);
    context.fillRect(200,200,200,100);
</script>
</body>
</html>
```

用浏览器打开 demo5-5，页面效果如图 5-20 所示。

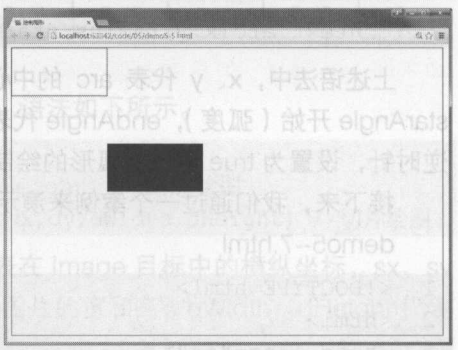


图5-20 demo5-5页面效果

在 demo5-5 中，通过坐标的不同绘制了两个不同位置的矩形边框和填充矩形。在 canvas 中还有一个相当于橡皮擦的方法，使用它可以清除矩形内绘制的内容，语法如下所示。

```
context.clearRect(x,y,width,height)
```

在上面的语法中，x、y 代表要清除矩形起点的横纵坐标，width 和 height 代表要清除矩形的宽和高，具体

用法如 demo5-6 所示。

demo5-6.html

```
<!DOCTYPE html>
<html>
<head lang="en">
    <meta charset="UTF-8">
    <title>清除矩形</title>
</head>
<body>
<canvas id="cavsElem">
    你的浏览器不支持 canvas，请升级浏览器
</canvas>
<script>
    //===============清除矩形====================
    //获得画布
    var canvas = document.getElementById('cavsElem');
    var context = canvas.getContext('2d');  //获得上下文
    //设置标签的属性宽高和边框
    canvas.width = 900;
    canvas.height = 600;
    canvas.style.border="1px solid #000";
    //绘制矩形
    context.strokeRect(0,0,200,100);
    context.fillRect(200,200,200,100);
    //清除矩形
    context.clearRect(100,50,200,200);
</script>
</body>
</html>
```

用浏览器打开 demo5-6，页面效果如图 5-21 所示。

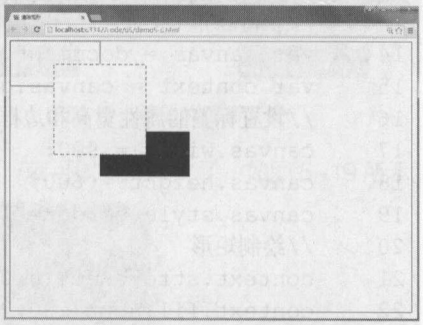


图5-21　demo5-6页面效果

在图 5-21 中，虚线区域为被清除的矩形区域，画布上的任意图形都可以用这样的方式来清除。

canvas 绘制圆形

在 canvas 中可以使用 arc()方法来绘制弧形和圆形，具体语法如下所示。

```
context.arc(x,y,radius,startAngle, endAngle,bAntiClockwise);
```

上述语法中，x、y 代表 arc 的中心点坐标；radius 代表圆形半径的长度；startAngle 代表以 starAngle 开始（弧度），endAngle 代表以 endAngle 结束（弧度）；bAntiClockwise 代表是否是逆时针，设置为 true 意味着弧形的绘制是逆时针方向的，false 则为顺时针。

接下来，我们通过一个案例来演示如何使用 arc()方法绘制圆形和弧形，如 demo5-7 所示。

demo5-7.html

```
<!DOCTYPE html>
<html>
<head lang="en">
    <meta charset="UTF-8">
```

```
    <title>绘制圆形和弧形</title>
</head>
<body>
<canvas id="cavsElem" width='400' height="300">
    你的浏览器不支持canvas，请升级浏览器
</canvas>
<script>
    /* 绘制圆形*/
    //获得画布并上下文对象
    var context = document.getElementById('cavsElem').getContext('2d');
    context.beginPath();//开始路径
    context.arc(100,100,100,0,2*Math.PI,true);//绘制圆形，true逆时针
    context.closePath();//关闭路径
    context.fillStyle = 'green';//设置填充颜色
    context.fill();//填充
    /* 绘制弧形*/
    context.beginPath();//开始路径
    context.strokeStyle = "#fff";//设置描边颜色
    context.lineWidth = 5;//设置线的粗细
    context.arc(100,150,25,Math.PI/6,5*Math.PI/6,false);//绘制弧形，false顺时针
    context.stroke();//描边
</script>
</body>
</html>
```

图5-22 demo5-7页面效果

用浏览器打开 demo5-7，页面效果如图 5-22 所示。

在 demo5-7 中，fillStyle()方法用于设置图形的填充颜色，strokeStyle()方法用于设置描边的颜色，lineWidth 属性用于设置线条的粗细（以像素为单位）。这些，样式设置同样可以应用于其他任意图形。arc()方法的参数中，bAntiClockwise 设置为 false，代表要绘制一个弧形，使用 Math.PI 来获取圆周率π的值，并且使用它来计算弧度值。特殊角度数和弧度数的对应如表 5-5 所示。

表 5-5 特殊角度数和弧度数的对应

度	0°	30°	45°	60°	90°	120°	135°	150°	180°	270°	360°
弧度	0	π/6	π/4	π/3	π/2	2π/3	3π/4	5π/6	π	3π/2	2π

canvas 绘制图片

canvas 中的绘制图片其实就是把一幅图放在画布上，语法如下所示。

```
context.drawImage(image, dx, dy) //绘制原图
context.drawImage(image, dx, dy, dWidth, dHeight) //缩放绘图
context.drawImage(image,sx,sy,sWidth,sHeigh,dx,dy,dWidth,dHeight) //切片绘图
```

在上述语法中，image 代表图片的来源，dx、dy 代表在 Image 目标中的横纵坐标，sx、sy 是 Image 在源中的起始坐标，sWidh、sHeight 代表源中图片的宽和高，dWidth、dHeight 代表目标的宽和高。

drawImag()方法的常用方式是绘制原图，如 demo5-8 所示。

demo5-8.html

```
<!DOCTYPE html>
<html>
<head lang="en">
    <meta charset="UTF-8">
    <title>绘制图片</title>
</head>
<body>
<canvas id="cavsElem" width="400" height="300" >
    你的浏览器不支持 canvas，请升级浏览器
</canvas>
<script type="text/javascript">
    //获得画布
    var canvas=document.getElementById('cavsElem');
    //设置画布边框
    canvas.style.border="1px solid #000";
    //获取上下文
    var context = canvas.getContext('2d');
    //创建图片对象
    var img=new Image();
    //设置图片路径
    img.src="images/demo5-8/draw.jpg";
    //当页面加载完成时使用此图片
    img.onload = function(){
        //使用 canvas 绘制图片
        context.drawImage(img,0,0);
    };
</script>
</body>
</html>
```

用浏览器打开 demo5-8，页面效果如图 5-23 所示。

图5-23　demo5-8页面效果

在 demo5-8 中必须使用图片对象的 onload 事件，否则是看不到运行效果的，因为绘制图片的基础是这个图片已经被加载了。

canvas 中的其他方法

canvas 中提供的有关图形绘制的方法还有很多，本书中不能一一列举，但是有必要介绍几个本项目涉及的方法，具体如下。

1. clip()方法

clip()方法用于从原始画布上剪切任意形状和尺寸的区域，具体使用方法如 demo5-9 所示。

demo5-9.html

```
<!DOCTYPE html>
<html>
<head lang="en">
    <meta charset="UTF-8">
```

```
    <title>clip()剪切任意形状和尺寸区域</title>
</head>
<body>
<canvas id="cavsElem" width="400" height="300" >
    你的浏览器不支持canvas，请升级浏览器
</canvas>
<script>
    //获得画布
    var canvas=document.getElementById('cavsElem');
    //设置画布边框
    canvas.style.border="1px solid #000";
    //获取上下文
    var context = canvas.getContext('2d');
    //剪切矩形区域
    context.rect(50,20,200,120);//(x,y,width,height)
    context.stroke();//描边
    context.clip();
    //在clip()之后绘制圆形，只有被剪切区域的内圆形可见
    context.arc(200,100,70,0,2*Math.PI,true);//(x，y，半径，开始弧度，结束弧度，
                                                true代表逆时针绘制圆形)
    context.fillStyle="pink";
    context.fill();//填充
</script>
</body>
</html>
```

用浏览器打开demo5-9，页面效果如图5-24所示。

图5-24 demo5-9页面效果

2. save()和restore()方法

在canvas绘制图形的过程中，有时网页需要多次显示相同的效果。例如，绘制圆形后绘制矩形，然后在触发某个事件时需要回到绘制圆形的状态,这时就用到了save()和restore()方法。save()用来保存画布的绘制状态；保存绘制一个圆形的状态，当绘制矩形后需要回到之前的状态，这时可以使用restore()方法。restore()方法用于移除自上一次调用save()方法所添加的任何效果。

【项目分析】

有了前导知识作为铺垫，接下来我们分析一下怎样完成该项目。该页面的页面标注和页面结构如图5-25和图5-26所示。

如图5-26所示，该页面分为照片部分和按钮部分，照片由一个外层 标签嵌套<img>标签和<canvas>标签构成。

该页面的实现细节具体分析如下。

（1）使用<a>标签制作两个按钮，并为按钮设置样式。

（2）图片模糊的效果使用CSS滤镜“filter: blur(px)”来实现，该属性可以用于实现类似于近视者忘了戴眼镜时看东西的模糊效果。

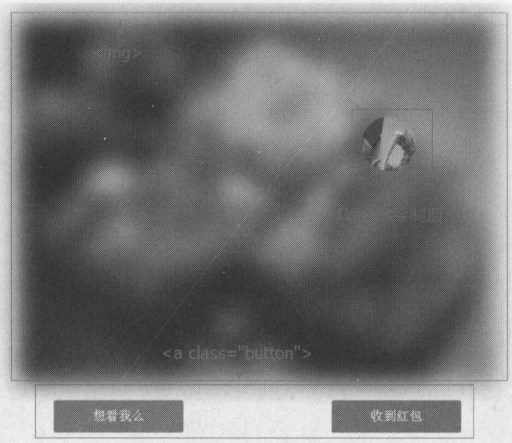


图5-25　页面标注

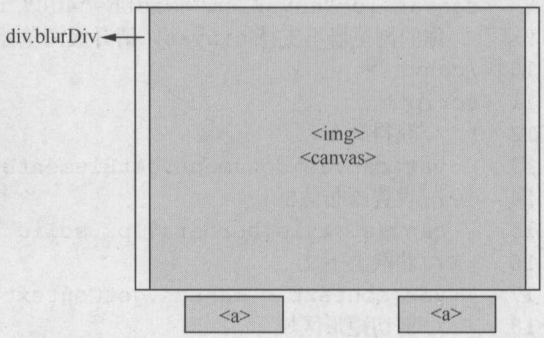


图5-26　页面结构

（3）各元素的定位都与 标签相对定位，需要注意的是，两个按钮要显示在最上层，z-index值最大，可以设置为“999”，其次是<canvas>标签，最后是<img>标签，圆形可显示的部分是使用 canvas 绘制圆形来实现的。

对页面的基本结构有所了解后，接下来我们分析一下 JavaScript 代码部分的实现，具体分析如下。

（1）绘制圆形：setRegion(Region)方法中需要使用 clip()方法剪切圆形区域，然后在圆形区域中绘制图片，就是图 5-24 中的圆形效果。

（2）绘制图片：draw(image)方法中需要调用 setRegion(Region)方法，并使用 clearRect()方法清除上一次绘制的圆形，保证只显示一个圆形区域。

（3）初始画布：initCanvas()方法中调用 draw(image)方法。

（4）“想看我么”按钮单击事件 reset()方法，在该方法中调用 initCanvas()方法，每次在不同的位置绘制圆形区域。

（5）“收到红包”按钮单击事件 show()方法，在该方法中调用 draw(image)方法，使圆形半径大于画布，这时就可以绘制完整的图片了，也就是收到红包的效果。

【代码实现】

接下来，即可开始编写代码来实现该项目。该项目的 HTML 页面代码如 code\05\0502\0502.html 所示。

0502.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>发红包才能看的照片</title>
<link rel="stylesheet" type="text/css" href="css/photo.css">
</head>
<body>
<div class="blurDiv">
    <img class="blurImg" id="blurImg" src="images/pic.jpg">
```

```
    <canvas id="myCanvas"></canvas>
   <a href="javascript:reset()" class="button" id="buttonReset">想看我么</a>
    <a href="javascript:show()" class="button" id="buttonShow">收到红包</a>
</div>
<script src="js/photo.js" type="text/javascript"></script>
</body>
</html>
```

该项目的 CSS 样式代码如 code\05\0502\css\ photo.css 所示。

photo.css

```
/*第 5 单元 项目 5-2 发红包才能看的照片*/
body {
  padding: 0;
  margin: 0;
}
.blurDiv {
  position: relative;
  width:877px;
  height: 672px;
  margin: 50px auto 0;/*外边距：顶部 50px 左右水平居中 底部 0*/
}
.blurDiv .blurImg {
  width: 870px;
  height: 672px;
  display: block;/*把行元素强制转换为块元素*/
  filter: blur(20px);/*滤镜：模糊程度 20px*/
  -webkit-filter: blur(20px);/*webkit 保留 否则 filter 没有作用*/
  position: absolute;
  top: 0;
  left: 0;
  z-index: 0;
}
.blurDiv #myCanvas {
  display: block;
  margin: 0 auto;
  z-index: 99;
  position: absolute;
  top: 0;
  left: 0;
}
 .blurDiv .button {
  display: block;
  width: 240px;
  height: 60px;
  border-radius: 5px;/*边框圆角 5px*/
  line-height: 60px;/*行高 60px*/
  text-align: center;/*文字水平居中*/
  position: absolute;
  top: 105%;
  font-family: arial;
```

```
    font-size: 1.5em;/*em是字体大小的单位，与px的区别在此不做详细介绍*/
    color: #fff;
    text-decoration: none;/*清除字体样式*/
    z-index: 999;
}
 /*给该元素设置绝对定位和背景颜色*/
.blurDiv #buttonReset {
    left: 7%;/*距离父元素左边7%*/
   background-color: #c86814;
}
/*当鼠标悬停在该元素时，背景颜色变为#ffb151*/
.blurDiv #buttonReset:hover {
   background-color: #ffb151;
}
.blurDiv #buttonShow {
    right: 7%;/*距离父元素右边7%*/
   background-color: #ff2f2e;
}
.blurDiv #buttonShow:hover {
   background-color: #ff596b;
}
```

该项目的 JavaScript 代码如 code\05\0502\js\ photo.js 所示。

photo.js

```
/*第5单元 项目5-2 发红包才能看的照片*/
var canvasWidth = 877;//声明画布的宽
var canvasHeight = 672;//声明画布的高
var canvas = document.getElementById("myCanvas");//获取画布
var context = canvas.getContext("2d");//获取画布的上下文
canvas.width = canvasWidth;
canvas.height = canvasHeight;
var image = new Image();//声明图片
var radius = 50;//声明半径
image.src = "images/pic.jpg";//获取图片路径
image.onload = function(e){
    initCanvas();//初始画布
}
function initCanvas(){
    Region =
          {x:Math.random()* (canvas.width-2*radius)+radius,y:Math.random()
        * (canvas.height-2*radius)+radius,r:radius}
    draw(Region);//绘制图片
}
//绘制圆形，用clip()方法剪切圆形区域
function setRegion(Region){
    context.beginPath();
    context.arc(Region.x,Region.y,Region.r,0,Math.PI*2,false);
    context.clip();
}
function draw(){
```

```
    //用于每次清除上一次绘制的圆形，保证只显示一个圆形区域
    context.clearRect(0,0,canvas.width, canvas.height);
    context.save();
    setRegion(Region);
    context.drawImage(image,0,0);
    context.restore();
}
//单击事件 reset()方法，在该方法中调用 initCanvas()方法，每次在不同的位置绘制圆形区域
function reset(){
    initCanvas();
}
//单击事件 show()方法，在该方法中调用 draw(image)方法，使圆形半径大于画布，这时就可以
  绘制完整的图片了，也就是收到红包的效果
function show(){
    Region.r = 2*Math.max(canvas.width,canvas.height);
    draw(image,Region);
}
```

【项目总结】

本项目的练习重点：

本项目主要练习的知识点是 canvas 绘制圆形和绘制图片以及 canvas 图形绘制中的常用方法 clip()、save()、restore()等。

本项目的练习方法：

建议读者在编码前，要了解整个项目的构成和效果。

（1）初始照片用 CSS 滤镜效果处理；用 canvas 绘制一个圆形可看清部分的图片。

（2）左按钮单击事件，可以随机绘制圆形。

（3）右按钮单击事件，改变圆形的半径，即可看见清晰的图片。

了解了项目要实现的效果后，再根据代码提示，编写各个功能。

【思考题】

1. 请简述什么是 canvas 以及使用 canvas 绘制图形的步骤。

2. 请简述使用 canvas 绘制的过程中 beginPath()和 closePath()的作用，这两种方法是否需要搭配使用。

Responsive Web Design

6 Chapter

单元 6
音频与视频

当前 Web 网站中，音频和视频早已成为网站重要的组成部分。但是长久以来，音频和视频一直依赖于第三方插件，插件会给网站带来一些性能和稳定性的问题。而<audio>与<video>标签的出现让音频与视频网站开发有了新的选择。<audio>与<video>标签用于播放音频和视频，并且 HTML5 规范为其提供了可脚本化控制的 API。本单元将就针对音频与视频标签的使用进行详细的讲解。

【教学导航】

学习目标	1. 掌握<video>标签和< audio>标签的使用 2. 掌握 HTML5 为 Video 和 Audio 对象提供的方法和事件 3. 复习 JavaScript 的运算符和流程结构语句
教学方式	本单元重点训练读者对<video>标签和< audio>标签的使用。需要注意的是： 项目 6-1 是为了让读者在不允许使用<video>标签的播放控件的情况下，也能够通过 JavaScript 实现对视频播放的控制； 项目 6-2 中用 for 循环语句实现对多个音频文件的控制； 项目 6-3 中要注意 div 的定位、显示与隐藏
重点知识	1. 浏览器对视频文件和音频文件的支持情况 2. HTML DOM Video 对象 3. HTML DOM Audio 对象
关键词	video、audio、source、HTML DOM Video、HTML DOM Audio、if、for、getElementsByTagName

【项目 6-1】 视频播放器

【项目描述】

视频播放器是一种用于播放各种视频文件的多媒体播放软件。本项目将带领读者完成一个自定义控制栏的 HTML5 网页视频播放器，如图 6-1 所示。

图6-1　视频播放器

【前导知识】

<video>标签的使用

在 HTML5 之前，网页中只能处理文字和图像数据，在 HTML5 中为网页提供了处理视频数

据的能力。使用<video>标签来定义视频播放器，其功能不仅仅是一个标签而已，<video>标签的控制栏，实现了包括播放、暂停、进度和音量控制、全屏等功能，更重要的是我们可以自定义这些功能和控制栏的样式。

视频可以理解为一系列连续的图片，<video>标签的使用方法与<img>标签非常相似，具体语法如下所示。

```
<video src="视频文件路径" controls>您的浏览器不支持 video 标签</video>
```

在上面的语法中，src 属性用于设置视频文件的路径，也可以为该标签设置 width 和 height 的值；controls 属性用于为视频提供播放控件，src 和 controls 是<video>标签的基本属性，并且<video>和</video>之间还可以插入文字，用于在浏览器不能支持时显示。

使用<img>标签时会涉及图片格式的问题，如 jpg、gif 等，视频文件也有不同的格式，<video>标签支持 3 种视频格式，具体如下。

- Ogg：带有 Theora 视频编码和 Vorbis 音频编码的 Ogg 文件。
- MPEG 4：带有 H.264 视频编码和 AAC 音频编码的 MPEG 4 文件。
- WebM：带有 VP8 视频编码和 Vorbis 音频编码的 WebM 文件。

浏览器对视频文件的支持情况如表 6-1 所示。

表 6-1　浏览器对视频文件的支持情况

视频格式	IE 9	Firefox 4.0	Opera 10.6	Chrome 6.0	Safari 3.0
Ogg		支持	支持	支持	
MPEG 4	支持			支持	支持
WebM		支持	支持	支持	

src 其实就是 source 的缩写，意为来源，这里指的是路径。从表 6-1 中可以看出，目前为止没有一种视频格式让所有浏览器都支持，为此，HTML5 中提供了<source>标签，用于指定多个备用的不同格式文件的路径，语法如下所示。

```
<video controls>
    <source src="视频文件地址" type="媒体文件类型/格式">
    <source src="视频文件地址" type="媒体文件类型/格式">
    ……
</video>
```

在上述语法中，媒体文件类型为 video 时代表视频，<source>标签对于音频文件同样适用，需要把 video 改成 audio。

对<video>标签有了基本的了解后，接下来我们通过一个案例来演示<video>标签的具体使用方法，如 demo6-1 所示。

demo6-1.html

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <meta charset="utf-8">
5  <title>video 元素</title>
6  </head>
7  <body>
```

```
8   <video src="video/movie.mp4">您的浏览器不支持 video 标签</video><br/><br/>
9   <video src="video/movie.mp4" controls>您的浏览器不支持 video 标签</video>
10 </body>
11 </html>
```

用浏览器打开 demo6-1，页面效果如图 6-2 所示。

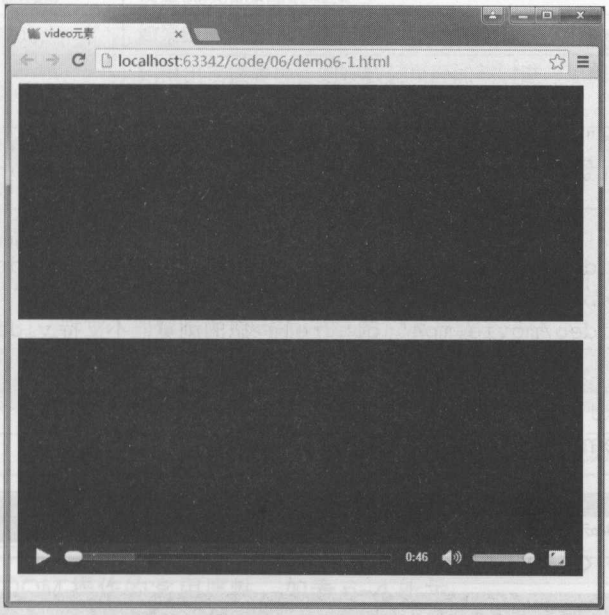


图6-2　demo6-1页面效果

在图 6-2 中，上面部分是<video>标签不添加 controls 属性的效果，controls 属性用于设置或返回浏览器应当显示标准的音视频控件。单击“▶”播放按钮，视频开始播放，如图 6-3 所示。

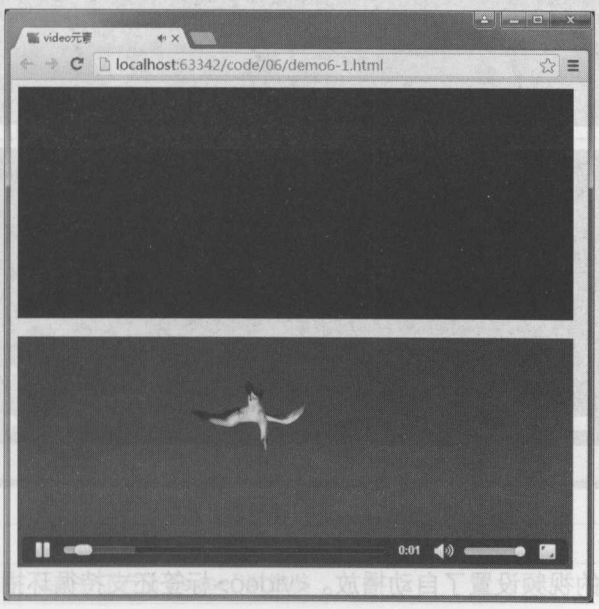


图6-3　视频开始播放的效果

图 6-3 播放中的视频下方为标准的音视频控件，包括 7 项功能：播放、暂停、进度条、音量、全屏切换（供视频）、字幕（当可用时）、轨道（当可用时）。没有音视频控件的情况下视频也是可以播放的，可以利用<video>标签的 autoplay 属性设置视频自动播放，如 demo6-2 所示。

demo6-2.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>video 元素</title>
</head>
<body>
<video src="video/movie.mp4" autoplay>您的浏览器不支持 video 标签
</video><br/><br/>
<video src="video/movie.mp4" controls>您的浏览器不支持 video 标签</video>
</body>
</html>
```

用浏览器打开 demo6-2，页面效果如图 6-4 所示。

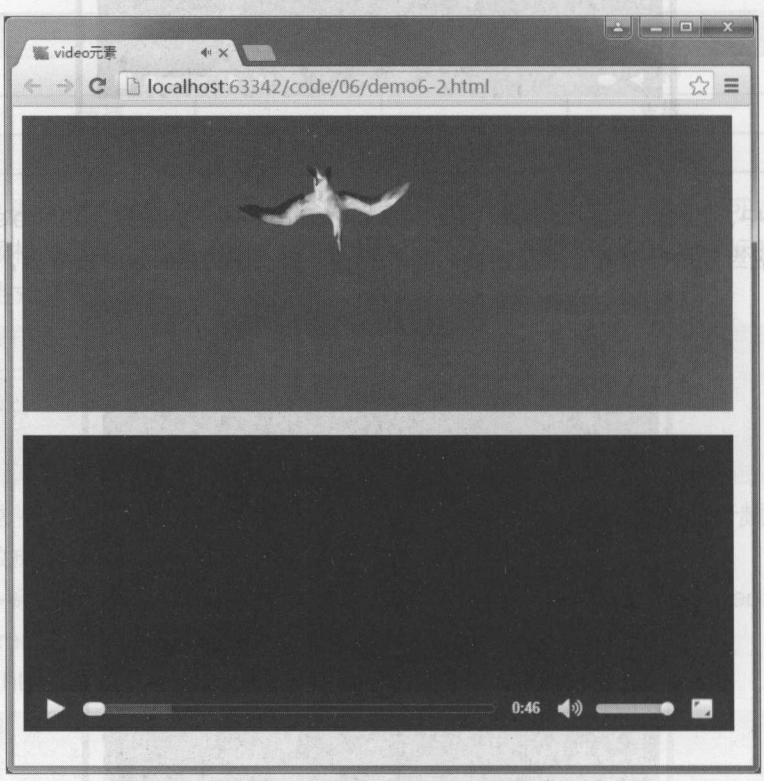


图6-4　demo6-2页面效果

在图 6-4 中上面的视频设置了自动播放。<video>标签还支持循环播放的功能，也是通过属性来控制的。<video>标签用于控制视频播放的常用属性，如表 6-2 所示。

表 6-2 <video>标签的常用属性

属性	允许取值	取值说明
autoplay	autoplay	如果出现该属性，则视频在就绪后马上播放
controls	controls	如果出现该属性，则向用户显示控件，如“播放”按钮
height	pixels	设置视频播放器的高度
loop	loop	如果出现该属性，则当媒体文件播放完后再次开始播放
preload	preload	如果出现该属性，则视频在页面加载时进行加载，并预备播放。如果使用“autoplay”，则忽略该属性
src	url	要播放的视频的 URL
width	pixels	设置视频播放器的宽度

HTML DOM Video 对象

HTML5 为 Video 对象提供了用于 DOM 操作的方法和事件，常用方法如表 6-3 所示。

表 6-3 Video 对象的常用方法

方法	描述
load()	加载媒体文件，为播放做准备，通常用于播放前的预加载，也用于重新加载媒体文件
play()	播放媒体文件。如果视频没有加载，则加载并播放；如果视频是暂停的，则变为播放
pause()	暂停播放媒体文件
canPlayType()	测试浏览器是否支持指定的媒体类型

Video 对象用于 DOM 操作的常用属性，如表 6-4 所示。

表 6-4 Video 对象的常用属性

属性	描述
currentSrc	返回当前视频的 URL
currentTime	设置或返回视频中的当前播放位置（以秒计）
duration	返回视频的长度（以秒计）
ended	返回视频的播放是否已结束
error	返回表示视频错误状态的 MediaError 对象
paused	设置或返回视频是否暂停
muted	设置或返回是否关闭声音
volume	设置或返回视频的音量
height	设置或返回视频的高度值
width	设置或返回视频的宽度值

Video 对象用于 DOM 操作的常用事件，如表 6-5 所示。

表 6-5 Video 对象的常用事件

事件	描述
play	当执行方法 play()时触发
playing	正在播放时触发
pause	当执行了方法 pause()时触发
timeupdate	当播放位置被改变时触发

续表

事件	描述
ended	当播放结束后停止播放时触发
waiting	在等待加载下一帧时触发
ratechange	在当前播放速率改变时触发
volumechange	在音量改变时触发
canplay	以当前播放速率，需要缓冲时触发
canplaythrough	以当前播放速率，不需要缓冲时触发
durationchange	在播放时长改变时触发
loadstart	在浏览器开始在网上寻找数据时触发
progress	在浏览器正在获取媒体文件时触发
suspend	在浏览器暂停获取媒体文件，且文件获取并没有正常结束时触发
abort	在中止获取媒体数据时触发，但这种中止不是由错误引起的
error	在获取媒体过程中出错时触发
emptied	在所在网络变为初始化状态时触发
stalled	在浏览器尝试获取媒体数据失败时触发
loadedmetadata	在加载完媒体元数据时触发
loadeddata	在加载完当前位置的媒体播放数据时触发
seeking	在浏览器请求数据时触发
seeked	在浏览器停止请求数据时触发

了解了 Video 对象的属性、方法和事件后，接下来我们通过一个案例来演示如何用 JavaScript 代码操作 Video 对象，具体使用方法如 demo6-3 所示。

demo6-3.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>JavaScript 操作 video 对象</title>
</head>
<body>
<video id="myVideo" src="video/myVideo.ogv">您的浏览器不支持 video 标签</video>
<!--<br/><br/>-->
<input  type="button" value="播放/暂停" onclick="playPause()"/>
</body>
<script>
    var myVideo=document.getElementById("myVideo");
    function playPause()
    {
        if (myVideo.paused)
            myVideo.play();
        else
            myVideo.pause();
```

```
20      }
21  </script>
22  </html>
```

用浏览器打开 demo6-3，页面效果如图 6-5 所示。

在 demo6-3 中定义了一个用于控制播放或者暂停的按钮，然后为该按钮的 onclick 事件定义方法 playPause ()。使用 JavaScript 中的 if 条件语句进行状态判断，当该播放器的状态为暂停（默认没有播放的视频会被识别为暂停状态）时调用 play()方法，切换为播放，单击“播放/暂停”按钮会切换到播放的状态，如图 6-6 所示。

图6-5 demo6-3页面效果

图6-6 播放的状态

再次单击“播放/暂停”按钮会切换到暂停的状态，JavaScript 的 if 条件语句在下面的小节中会进行详细介绍。

JavaScript 运算符和 if 条件语句

1. 运算符

运算符是程序执行特定算术或操作的符号，用于执行程序代码运算。JavaScript 中的运算符主要包括算术运算符、比较运算符、赋值运算符、逻辑运算符和条件运算符 5 种，具体介绍如下。

（1）算术运算符

算术运算符用于链接运算表达式，主要包括加（+）、减（-）、乘（*）、除（/）、取模（%）、自增（++）、自减（--）等运算符，如表 6-6 所示。

表 6-6 算术运算符

算术运算符	描述
+	加运算符
-	减运算符
*	乘运算符
/	除运算符
++	自增运算符，该运算符有i++（在使用i之后，使i的值加1）和++i（在使用i之前，先使i的值加1）两种
--	自减运算符，该运算符有i--（在使用i之后，使i的值减1）和--i（在使用i之前，先使i的值减1）两种

（2）比较运算符

比较运算符在逻辑语句中使用，用于判断变量或值是否相等，返回一个布尔类型的值，true

或 false。常用的比较运算符如表 6-7 所示。

表 6-7　比较运算符

比较运算符	描述
<	小于
>	大于
<=	小于等于
>=	大于等于
= =	等于，只根据表面值进行判断，不涉及数据类型。例如，“27”= =27 的值为 true
!=	不等于，只根据表面值进行判断，不涉及数据类型。例如，“27”!=27 的值为 false

（3）逻辑运算符

逻辑运算符是根据表达式的值来返回真值或假值，常用的逻辑运算符如表 6-8 所示。

表 6-8　逻辑运算符

逻辑运算符	描述
&&	逻辑与，只有当两个操作数 a、b 的值都为 true 时，a&&b 的值才为 true；否则为 false
||	逻辑或，只有当两个操作数 a、b 的值都为 false 时，a||b 的值才为 false；否则为 true
!	逻辑非，!true 的值为 false，而!false 的值为 true

（4）赋值运算符

最基本的赋值运算符是等于号“=”，用于对变量进行赋值。其他运算符可以和赋值运算符联合使用，构成组合赋值运算符，如表 6-9 所示。

表 6-9　赋值运算符

赋值运算符	描述
=	例如，username=“name”
+ =	例如，a+=b，相当于 a=a+b
- =	例如，a-=b，相当于 a=a-b
*=	例如，a*=b，相当于 a=a*b
/ =	例如，a/=b，相当于 a=a/b
% =	例如，a%=b，相当于 a=a%b

（5）条件运算符

条件运算符是 JavaScript 中的一种特殊的三目运算符，其语法格式如下。

```
操作数? 结果 1: 结果 2
```

如果操作数的值为 true，则整个表达式的结果为“结果 1”，否则为“结果 2”。

2. 条件语句

所谓条件语句就是对语句中不同条件的值进行判断，进而根据不同的条件执行不同的语句。if 判断语句是最基本、最常用的条件控制语句，通过判断条件表达式的值为 true 或者 false 来确定是否执行某一条语句，主要包括单向判断语句、双向判断语句和多向判断语句，具体讲解如下。

（1）单向判断语句

单向判断语句是结构最简单的条件语句，如果程序中存在绝对不执行某些指令的情况，就可以使用单向判断语句，其语法格式如下。

```
if（执行条件）{
    执行语句
}
```

在上面的语法结构中，if 可以理解为“如果”，小括号“()”内用于指定 if 语句中的执行条件，大括号“{}”内用于指定满足执行条件后需要执行的语句。

（2）双向判断语句

双向判断语句是 if 条件语句的基础形式，只是在单向判断语句的基础上增加了一个从句，其基本语法格式如下。

```
if（执行条件）{
执行语句 1
}else{
执行语句 2
}
```

双向判断语句的语法格式和单向判断语句类似，只是在其基础上增加了一个 else 从句，表示如果条件成立则执行“语句 1”，否则执行“语句 2”。

（3）多向判断语句

多向判断语句是根据表达式的结果判断一个条件，然后根据返回值做进一步的判断，其基本语法格式如下。

```
if（执行条件 1）{
执行语句 1
}
else if（执行条件 2）{
执行语句 2
}
else if（执行条件 3）{
执行语句 3
}
……
```

在多向判断语句的语法中，通过 else if 语句可以对多个条件进行判断，并且根据判断的结果执行相关的语句。

【项目分析】

有了前导知识作为铺垫，接下来我们分析一下如何实现 HTML5 视频播放器。该页面的页面标注和页面结构如图 6-7 和图 6-8 所示。

如图 6-8 所示，该页面由一个<video>标签、多个<source>标签和多个<button>标签构成。

该页面的实现细节具体分析如下。

（1）这里的<source>标签分别引用 3 种格式的视频文件，即“ogg”“webm”和“mp4”格式。

由于谷歌浏览器和火狐浏览器都支持前两种格式，而 IE 浏览器只支持“mp4”格式，所以这里引用的“mp4”格式的视频与前两个视频内容不同，这样方便效果的查看。例如，图 6-7 中所示的页面是用谷歌浏览器打开的，用 IE 浏览器播放的效果是另外一段“mp4”格式的视频，如图 6-9 所示。

图6-7　页面标注

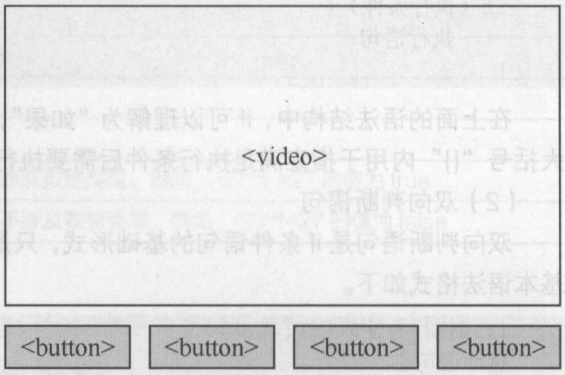


图6-8　页面结构

图6-9　IE浏览器页面效果

（2）对所有按钮设置了统一的样式，包括背景色、字体、圆角边框、阴影等。

（3）当鼠标悬停在某个按钮上时应该变为小手的形状。

清楚了页面的基本结构后，接下来我们要做的就是使用 JavaScript 代码为按钮添加相应的功能，具体分析如下。

（1）控制视频播放、暂停：playPause()，在该方法中需要判断 Video 对象的状态。如果 Video 对象的状态为 paused，则调用 play()方法，并设置按钮文字为“暂停”；否则调用 pause()方法，并设置按钮文字为“播放”。

（2）控制视频快进、快退：goBack(val)，在该方法中通过控制 video 对象中 currentTime 的值来实现效果，currentTime+=val 等价于 currentTime=currentTime+val。

（3）控制视频音量：volume(val)，在该方法中通过控制 Video 对象的 volume 属性值，实现效果，可以参考 goBack()方法的实现。

（4）控制视频是否静音：isMuted()，在该方法中需要判断 Video 对象的 muted 状态，具体实现可以参考 playPause()方法。

【代码实现】

接下来，即可开始编写代码来实现该项目。该项目的 HTML 页面代码如 code\06\0601\0601.html 所示。

0601.html

```
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
    <title>HTML5 视频播放器</title>
    <link rel="stylesheet" type="text/css" href="css/video.css" />
</head>
<body >
<video id="myVideo"  width="630">
    <!--判断是否支持其播放器-->
    <source src="video/myVideo.ogg" type="video/ogg"/>
    <source src="video/myVideo.webm" type="video/webm"/>
    <source src="video/myVideo.mp4" type="video/mp4"/>
    您的浏览器不支持 video 标签
</video>
<br />
<button id="playPause" onclick="playPause();">播放</button>
<button onclick="goBack(5);">快进 5 秒</button>
<button onclick="goBack(-5);">快退 5 秒</button>
<button onclick="volume(0.1);">音量+</button>
<button onclick="volume(-0.1);">音量-</button>
<button id="isMuted" onclick="isMuted();">静音</button>
</body>
<script type="text/javascript" src="js/video.js" charset="gbk"></script>
</html>
```

该项目的 CSS 样式代码如 code\06\0601\css\ video.css 所示。

video.css

```
/*第 6 单元 项目 6-1 视频播放器 css 文件*/
/*设置所有按钮的样式*/
button {
    display: inline-block;/*强制转换为行内块元素*/
    outline: none;/*该元素获得焦点时，不出现虚线框（或高亮框）*/
    cursor: pointer;/*鼠标悬停时变为“小手”形状*/
    text-align: center;
    text-decoration: none;
    /*字体样式：字体大小 14px/行高 100% 字体名称*/
    font: 14px/100% Arial, Helvetica, sans-serif;
    padding:0 .5em 2em 0.55em;/*顶部内边距 0.5em、左右内边距 2em、底部内边距 0.55em*/
    text-shadow: 0 1px 1px rgba(0,0,0,.3);/*文本阴影*/
    -webkit-border-radius: 0.5em;
    -moz-border-radius: 0.5em;
    border-radius: 0.5em;/*边框圆角*/
    background-color: #944bc8;
```

```
    color: white;
}
/*设置所有按钮鼠标悬停效果*/
button:hover {
    text-decoration: none;/*清除文本样式*/
}
```

该项目的 JavaScript 代码如 code\06\0601\js\ video.js 所示。

video.js

```
/*第 6 单元 项目 6-1 视频播放器 js 文件*/
//获取 video 对象
var myVideo = document.getElementById("myVideo");
//控制播放/暂停的方法
function playPause()
{
    var ppButton=document.getElementById("playPause");
    if (myVideo.paused){
        myVideo.play();
        ppButton.innerHTML="暂停";
    }
    else{
        myVideo.pause();
        ppButton.innerHTML="播放";
    }
}
//控制快进/快退的方法
function goBack(val){
    myVideo.currentTime += val;
}
//控制音量的方法
function volume(val){
        myVideo.volume+=val;
}
//控制是否静音的方法
function isMuted(){
    var imButton=document.getElementById("isMuted");
    if (myVideo.muted){
        myVideo.muted=false;
        imButton.innerHTML="静音";
    }
    else{
        myVideo.muted=true;
        imButton.innerHTML="声音";
    }
}
```

【项目总结】

本项目的练习重点：

本项目主要练习的知识点是 HTML5 的<video>标签、HTML DOM Video 对象、JavaScript 运

算符和 if 条件语句的应用。

本项目的练习方法：

建议读者在编码时，先完成页面效果（HTML 和 CSS 代码），再逐个给每个按钮添加事件，控制视频播放。

本项目的注意事项：

本项目的 CSS 样式和 JavaScript 代码都使用了链入式的引入方式，需要注意的是，要在引入 JavaScript 文件的语句中加入 charset="gbk"，否则文件中的汉字会出现乱码问题。

【项目 6-2】 HTML5 Web 钢琴

【项目描述】

目前，网络上有许多支持在线弹奏的钢琴小游戏，可以直接在手机或 PC 端进行弹奏，效果像极了电子琴。本项目将带领读者完成一个 HTML5 的 Web 钢琴，如图 6-10 所示。

在图 6-10 中，当鼠标指针移动到某个琴键上时，鼠标指针变为“小手”形状，琴键颜色发生变化，如图 6-11 所示。

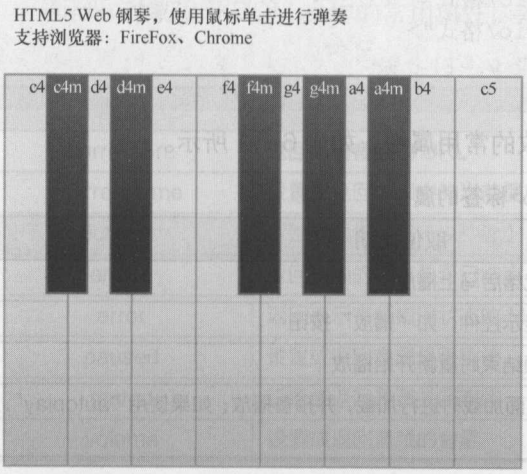


图6-10　HTML5 Web钢琴

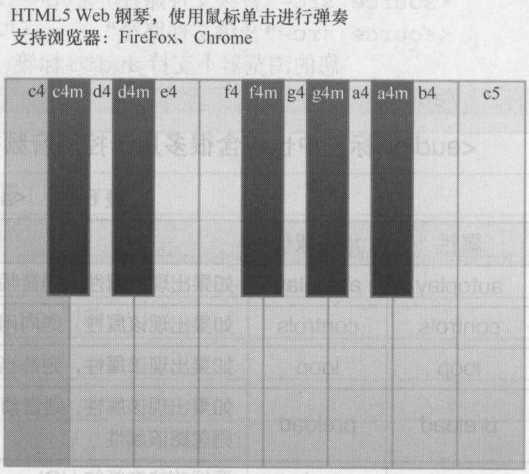


图6-11　鼠标移动的琴键效果

单击钢琴琴键，即可进行弹奏，琴键上的字符 c4、c4m 等为当前琴键的音调标识。

【前导知识】

<audio>标签的使用

目前为止，在网页中播放音频没有固定的标准，大多数音频是通过插件（如 Flash）来播放的，但并非所有浏览器都有同样的插件。HTML5 中提供了<audio>标签来定义 Web 上的声音文件或音频流，其使用方法与<video>标签基本相同，语法如下所示。

```
<audio src="音频文件路径" controls>您的浏览器不支持 audio 标签</audio>
```

<audio>标签同样支持引入多个音频源。多个音频源涉及音频的格式问题，当前<audio>标

签支持 3 种格式，具体如下。

• Vorbis：是类似于 ACC 的另一种免费、开源的音频编码，是用于替代 MP3 的下一代音频压缩技术。

• MP3：是一种音频压缩技术，其全称是动态影像专家压缩标准音频层面 3（Moving Picture Experts Group Audio Layer III），简称为 MP3。它被用来大幅度地降低音频数据量。

• Wav：是录音时用的标准的 Windows 文件格式，文件的扩展名为“WAV”，数据本身的格式为 PCM 或压缩型，属于无损音乐格式的一种。

与视频的支持情况相似，目前没有一种浏览器支持所有的音频格式，具体如表 6-10 所示。

表 6-10　浏览器对音频格式的支持情况

音频格式	IE 9	Firefox 4.0	Opera 10.6	Chrome 6.0	Safari 3.0
Ogg Vorbis		支持	支持	支持	
MP3	支持			支持	支持
Wav		支持	支持		支持

多个音频源使用<source>标签来定义，语法如下所示。

```
<audio controls>
  <source src="音频文件路径" type="audio/格式">
  <source src="音频文件路径" type="audio/格式">
        您的浏览器不支持 audio 标签
</audio>
```

<audio>标签中也包含很多用于控制音频播放的常用属性，如表 6-11 所示。

表 6-11　<audio>标签的属性

属性	允许取值	取值说明
autoplay	autoplay	如果出现该属性，则音频在就绪后马上播放
controls	controls	如果出现该属性，则向用户显示控件，如“播放”按钮
loop	loop	如果出现该属性，则每当音频结束时重新开始播放
preload	preload	如果出现该属性，则音频在页面加载时进行加载，并预备播放；如果使用“autoplay”，则忽略该属性
src	url	要播放的音频的 URL

从表 6-11 中可以看出，与<video>标签相比，<audio>标签没有 width 和 height 属性，其他属性名称都相同。

<audio>标签的具体用法如 demo6-4 所示。

demo6-4.html

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8">
5      <title>audio 标签的使用</title>
6  </head>
7  <body>
```

```
8  <audio src="audio/demo6-4/music.mp3" controls></audio>
9  </body>
10 </html>
```

用浏览器打开 demo6-4，页面效果如图 6-12 所示。

图6-12　demo6-4页面效果

图 6-12 所示的音频播放器的效果类似于视频播放器的播放控件，在不添加 controls 属性的情况下页面应该是空白，可以通过 JavaScript 控制音频的播放。

HTML DOM Audio 对象

HTML5 为 Audio 对象提供了用于 DOM 操作的方法和事件，常用方法如表 6-12 所示。

表 6-12　Audio 对象的常用方法

方法	描　述
load()	加载媒体文件，为播放做准备，通常用于播放前的预加载，也用于重新加载媒体文件
play()	播放媒体文件。如果音频没有加载，则加载并播放；如果音频是暂停的，则变为播放
pause()	暂停播放媒体文件
canPlayType()	测试浏览器是否支持指定的媒体类型

Audio 对象用于 DOM 操作的常用属性，如表 6-13 所示。

表 6-13　Audio 对象的常用属性

属性	描　述
currentSrc	返回当前音频的 URL
currentTime	设置或返回音频中的当前播放位置（以秒计）
duration	返回音频的长度（以秒计）
ended	返回音频的播放是否已结束
error	返回表示音频错误状态的 MediaError 对象
paused	设置或返回音频是否暂停
muted	设置或返回是否关闭声音
volume	设置或返回音频的音量

Audio 对象用于 DOM 操作的常用事件，如表 6-14 所示。

表 6-14　Audio 对象的常用事件

事件	描　述
play	当执行方法 play()时触发
playing	正在播放时触发
pause	当执行方法 pause()时触发
timeupdate	当播放位置被改变时触发
ended	当播放结束后停止播放时触发
waiting	在等待加载下一帧时触发
ratechange	在当前播放速率改变时触发

续表

事件	描　述
volumechange	在音量改变时触发
canplay	以当前播放速率，需要缓冲时触发
canplaythrough	以当前播放速率，不需要缓冲时触发
durationchange	在播放时长改变时触发
loadstart	在浏览器开始在网上寻找数据时触发
progress	在浏览器正在获取媒体文件时触发
suspend	在浏览器暂停获取媒体文件，且文件获取并没有正常结束时触发
abort	在中止获取媒体数据时触发，但这种中止不是由错误引起的
error	在获取媒体过程中出错时触发
emptied	在所在网络变为初始化状态时触发
stalled	在浏览器尝试获取媒体数据失败时触发
loadedmetadata	在加载完媒体元数据时触发
loadeddata	在加载完当前位置的媒体播放数据时触发
seeking	在浏览器请求数据时触发
seeked	在浏览器停止请求数据时触发

以上方法、属性和事件可以通过 JavaScript 来操作，用法与 Video 对象中的方法、属性等非常相似。例如，使用按钮来控制音频的播放，如 demo6-5 所示。

demo6-5.html

```
<!Doctype html>
<html>
<head>
    <meta charset="utf-8">
    <title>JavaScript 操作 audio 对象</title>
</head>
<script>
    //页面加载完毕后执行
    window.onload=function(){
        //通过标签名获取 button 按钮
        document.getElementsByTagName("button")[0].onclick=function(){
          //通过标签名获取 audio 对象
           document.getElementsByTagName("audio")[0].load();
           document.getElementsByTagName("audio")[0].play();
        }
    }
</script>
<body>
<audio src="audio/demo6-5/music.mp3"></audio>
<button >播放音乐</button>
</body>
</html>
```

用浏览器打开 demo6-5，页面效果如图 6-13 所示。

在 demo6-5 中使用标签名来获取某个标签时，默认得到的是数组对象。数组对象的下标从 0 开始，这里每种标签只有一个，所以使用下标 0 来获取对象。单击图 6-13 中所示的“播放音乐”按钮，音乐开始播放。

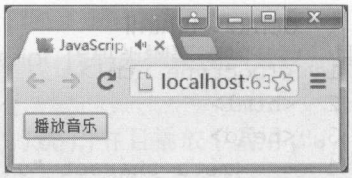
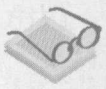


图6-13　demo6-5页面效果



多学一招：深入理解 Audio 和 Video 对象

其实，<audio>标签和<video>标签有很大的相似性，Audio 对象和 Video 对象的 DOM 操作功能其实都是由 HTMLMediaElement 对象统一定义的核心功能。Audio 对象指的是 HTMLAudioElement 对象，它完全继承了 HTMLMediaElement 对象提供的功能；而 Video 对象指的是 HTMLVideoElement 对象，在该对象中提供了额外的功能，主要表现在一些额外的属性上，如表 6-15 所示。

表 6-15　HTMLVideoElement 对象定义的额外属性

属性	描述
poster	获取或设置 poster 属性值
videoHeight	获取视频的原始高度
videoWidth	获取视频的原始宽度
height	设置或返回视频的高度值
width	设置或返回视频的宽度值

以上属性是 Audio 对象所没有的。

JavaScript 循环语句

循环语句用于批量操作，JavaScript 支持不同类型的循环，如下所示。

- for：循环代码块一定的次数。
- for/in：循环遍历对象的属性。
- while：当指定的条件为 true 时循环指定的代码块。
- do/while：同样当指定的条件为 true 时循环指定的代码块。

在上述 4 种循环语句中，最常用的就是 for 循环语句，语法如下所示。

```
for (语句 1; 语句 2; 语句 3)
  {
  被执行的代码块
  }
```

在上述语法中，语句 1 用于在循环开始之前设置变量（如 i=1）；语句 2 用于定义循环运行的条件（如 i 必须小于 3）；语句 3 用于在每次代码块已被执行后增加或减少一个值（如 i++），没有满足的条件时循环结束。

例如，正常情况下使用 JavaScript 输出数字代码如下。

```
document.write(0 + "<br>");
document.write(1 + "<br>");
document.write(2 + "<br>");
document.write(3 + "<br>");
document.write(4 + "<br>");
```

使用 for 循环语句可以起到简化代码的作用，如 demo6-6 所示。

demo6-6.html

```
<!Doctype html>
<html>
<head>
    <meta charset="utf-8">
    <title>JavaScript 操作 audio 对象</title>
</head>
<script>
    window.onload=function(){
        //循环输出 i 的值
        for(var i=0;i<=4;i++){
            document.write(i+"<br/>")
        }
    }
</script>
<body>
</body>
</html>
```

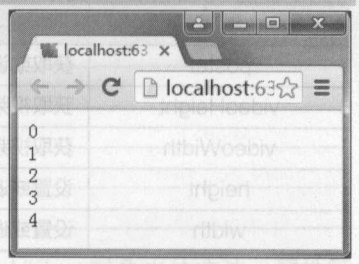


图6-14 demo6-6页面效果

用浏览器打开 demo6-6，页面效果如图 6-14 所示。

【项目分析】

有了前导知识作为铺垫，接下来我们分析如何实现 HTML5 Web 钢琴。该页面的页面标注和页面结构如图 6-15 和图 6-16 所示。

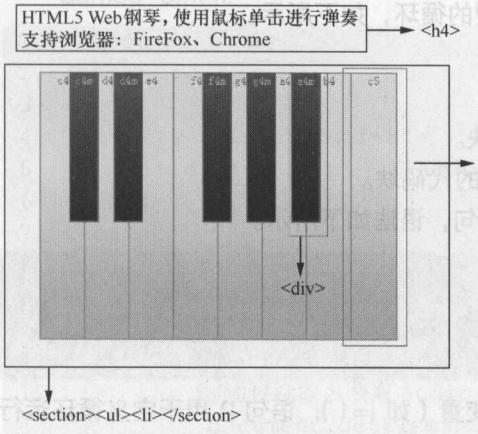


图6-15 页面标注

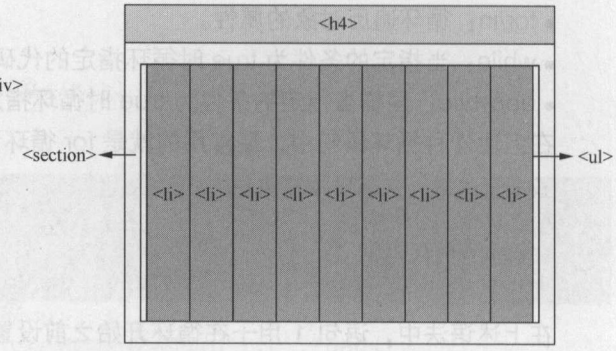


图6-16 页面结构

如图 6-16 所示的钢琴页面由标题部分和钢琴主体部分构成，两部分嵌套在一个<section>标签中。

该页面的实现细节具体分析如下。

（1）使用<h4>标签设置标题说明。

（2）使用<ul>列表嵌套多个 标签构成每个琴键，分别对两种琴键的 标签设置背景图片，该背景图片具有透明效果，当鼠标悬停到某个琴键上时，改变 的背景色。

（3）使用<audio>标签来定义每个琴键对应的音频文件，这里全部使用“ogg”格式，支持

谷歌浏览器和火狐浏览器。

清楚了该页面的结构后，接下来我们要做的就是使用 JavaScript 代码为钢琴添加音效，具体分析如下。

（1）当用鼠标单击某个琴键时，加载相应的音频（调用 load()方法），并且播放（调用 play()方法）。

（2）每个琴键（div）包括黑键都有对应的音频文件，也就是说有多少个 标签就有多少个<audio>标签。

（3）使用 document.getElementsByTagName(tagName)方法来获取琴键和对应音频文件的数组，然后通过数组的下标获取每个琴键和每个音频，数组的下标从 0 开始，然后是 1，2，3，4，…，*n*，使用 for 循环来循环调用音频的 load()和 play()方法。

【代码实现】

对项目的结构和样式有所了解后，即可开始编写代码来实现它。该项目的 HTML 页面代码如 code\06\0602\0602.html 所示。

0602.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>HTML5 Web 钢琴</title>
    <link rel="stylesheet" type="text/css" href="css/piano.css" />
    <script type="text/javascript" src="js/piano.js"></script>
</head>
<body>
<section class="main">
    <h4>HTML5 Web 钢琴，使用鼠标单击进行弹奏 <br/>支持浏览器：FireFox、Chrome</h4>
    <!--用<audio></audio>标签引入音频源-->
    <audio src="audio/c4.ogg"></audio>
    <audio src="audio/c4m.ogg"></audio>
    <audio src="audio/d4.ogg"></audio>
    <audio src="audio/d4m.ogg"></audio>
    <audio src="audio/e4.ogg"></audio>
    <audio src="audio/f4.ogg"></audio>
    <audio src="audio/f4m.ogg"></audio>
    <audio src="audio/g4.ogg"></audio>
    <audio src="audio/g4m.ogg"></audio>
    <audio src="audio/a4.ogg"></audio>
    <audio src="audio/a4m.ogg"></audio>
    <audio src="audio/b4.ogg"></audio>
    <audio src="audio/c5.ogg"></audio>
    <!--整个钢琴用 <ul></ul> 标签，每一个 <div></div> 标签对应一个琴键-->
    <ul>
        <li>
            <div class="key whiteKey">c4</div>
            <div class="key blackKey">c4m</div>
      </li>
```

```
        <li>
            <div  class="key whiteKey">d4</div>
            <div  class="key blackKey">d4m</div>
        </li>
        <li><div  class="key whiteKey">e4</div></li>
        <li>
            <div  class="key whiteKey">f4</div>
            <div class="key blackKey">f4m</div>
        </li>
        <li>
            <div class="key whiteKey">g4</div>
            <div class="key blackKey">g4m</div>
        </li>
        <li>
            <div  class="key whiteKey">a4</div>
            <div  class="key blackKey">a4m</div>
        </li>
        <li><div class="key whiteKey">b4</div></li>
        <li class="nobd"><div class="key whiteKey">c5</div></li>
    </ul>
    </section>
</body>
</html>
```

该项目的 CSS 样式代码如 code\06\0602\css\piano.css 所示。

piano.css

```
/*第 6 单元 项目 6-2 HTML5 Web 钢琴 css 文件*/
/* 给所有标签清除内边距、外边距、默认样式和边框*/
*{ padding:0; margin:0 ; list-style:none; border:0;}
.main {
    margin:0 auto;
    width:600px;
    height:420px;
    position:relative;
    top:150px;
}
ul li{
    width:60px;
    height:360px;
    float:left;/*向左浮动*/
    position:relative;
}
div.key {
    position: absolute;
    float:left;
    top:40px;
    text-align: center;
    font-size:16px;
```

```
    background-position: bottom left;/*背景位置：垂直方向在底部，水平方向在左边*/
    background-repeat: repeat-x;/*背景水平方向平铺*/

}
/*当鼠标悬停在该元素上时设置背景颜色为#cacbd1，鼠标变为“小手”形状*/
.key:hover{
    background-color: #cacbd1;
    cursor:pointer;
}
div.whiteKey {
    width:60px;
    height:360px;
    color:#666;
    background-color:#eee;
    border: 2px solid #999;
    background-image:url("../images/key_back_white.png");
}
div.blackKey {
    width:36px;
    height:200px;
    color:#ccc;
    margin-left: 40px;
    z-index: 999;/*z-index 属性值越大，该元素层离用户越近*/
    background-color:#666;
    border: 2px solid #666;
    background-image:url("../images/key_back_black.png");
}
```

该项目的 JavaScript 代码如 code\06\0602\js\piano.js 所示。

piano.js

```
/*第 6 单元 项目 6-2 HTML5 Web 钢琴 js 文件*/
//页面加载完毕后执行
window.onload = function(){
    //获取所有的 div 琴键
    var mydivs = document.getElementsByTagName("div");
    //获取所有的 audio 音频源
    var myaudios = document.getElementsByTagName("audio");
    //通过数组的下标获取每个琴键和每个音频，数组的下标从 0 开始
    //使用 for 循环来循环调用音频的 load()和 play()方法
    for(var i = 0 ; i <14; i++){
        mydivs[i].index = i;
   //当用鼠标单击某个琴键时，加载相应的音频（调用 load()方法），并且播放（调用 play()方法）
        mydivs[i].onclick = function(){
            myaudios[this.index].load(); //保持余音绕梁，开启新的声音
            myaudios[this.index].play();
        }
    }
}
```

【项目总结】

本项目的练习重点：

本项目主要练习的知识点是 HTML5 的<audio>标签、HTML DOM Audio 对象和 JavaScript 循环语句的应用。

本项目的练习方法：

建议读者在编码时，先完成页面效果（HTML 和 CSS 代码），再编写循环语句控制音频源的播放部分。

本项目的注意事项：

本项目的完成过程中需要注意音频文件与相应琴键的对应顺序，需要检查每个琴键的音调是否正确，如两个琴键发出相同声音就是不正确的，也可也通过其他播放器播放音频文件进行对比。

【项目 6-3】 音乐播放器

【项目描述】

音乐播放器是一种用于播放各种音频文件的多媒体播放软件，本项目将带领读者完成一款网页版的音乐播放器页面，如图 6-17 所示。

单击图 6-17 中的“播放”按钮“ ”，音乐开始播放，并显示“暂停”按钮。当鼠标指针悬停在某个按钮上时，按钮颜色将发生变化，如图 6-18 所示。

图6-17 音乐播放器

图6-18 音乐播放状态

在图 6-18 中的“播放”按钮下方，可以调节音量，如图 6-19 所示。

单击图 6-19 中的“ ”按钮，可以显示该歌曲的歌词，如图 6-20 所示。

单击图 6-20 中的“ ”按钮，即可收起歌词。

图6-19　调节音量

图6-20　显示歌词

【项目分析】

1. 页面结构

从音乐播放器的效果可以看出，该页面由背景图、按钮区域、信息区域、歌词区域等构成，页面标注和页面结构如图 6-21 和图 6-22 所示。

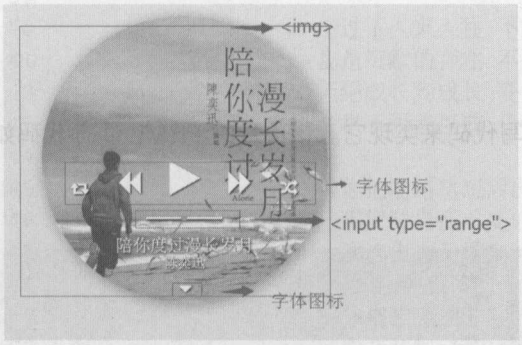


图6-21　页面标注

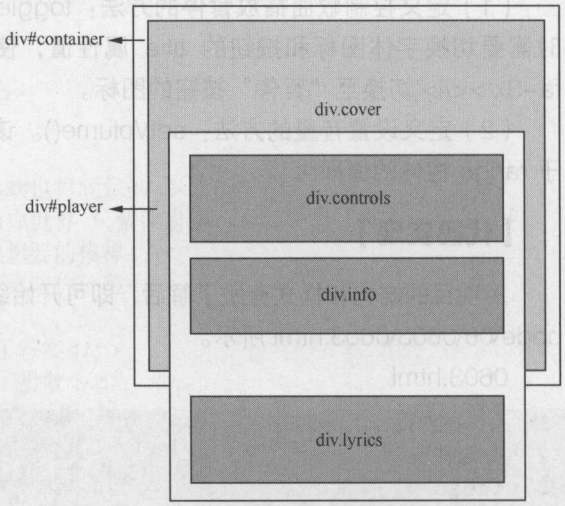


图6-22　页面结构

如图 6-22 所示，该页面主要由 div#container、div#player 和 div.cover 3 部分构成。

该页面的实现细节具体分析如下。

（1）id 值为 container 的 标签，用于整体页面布局。

（2）id 值为 player 的 标签中嵌入<audio>、<img>标签，用于定义音频和背景图片。

（3）在 div.cover 中添加 3 个 标签，分别用于放置控制按钮（div.controls），歌曲信息（div.info）和歌词（div.lyrics）。

（4）使用<label>嵌套<i>标签制作显示歌词的按钮，并在下方定义复选框。<label>标签中

使用 for 属性，单击<label>可以选中复选框，用于判断当按钮被选中时显示歌词。

（5）div.controls 的按钮区域可以使用<button>嵌套标签来实现，div.info 中音量控件使用<input type="range">控件，div.lyrics 歌词区域可以使用多个<p>标签定义。

2. 页面样式

清楚了页面的主要结构后，接下来我们进行样式分析，具体如下。

（1）通过 div+css 对页面进行整体控制，需要设置宽度、高度、绝对定位等样式。

（2）div#player 中需要设置圆角边框、阴影效果、背景图等，背景图设置了透明度。

（3）对 div.cover 设置上外边距，来隐藏歌词区域；设置过渡，实现显示歌词时的滑动效果。

（4）对 div.controls 设置相对定位、背景色、宽度等，对 div.controls 中的每个按钮设置字体图标、背景色、外边距、鼠标悬停效果等。

（5）对 div.info 中的歌曲信息和音量控件设置样式，使用伪元素::-webkit-slider-thumb 来改变 range 的默认样式，并设置鼠标悬停的效果。

（6）设置 div.lyrics 歌词区域样式，包括相对定位、宽高、背景色、字体大小、鼠标悬停效果等。

（7）设置显示歌词的按钮样式，使用字体图标，并使用:checked 选择器匹配已被选中的 checkbox，并设置被选中时 div.cover 以及其包含的 3 个 标签全部向上移动，达到显示歌词的效果。

3. JavaScript 脚本

设置了样式之后，接下来我们要做的就是使用 JavaScript 代码为播放器添加功能，具体分析如下。

（1）定义控制歌曲播放暂停的方法：togglePlayPause()。该方法中，在切换播放暂停的同时需要切换字体图标和按钮的 title 属性值，使用 obj.innerHTML = ''切换至“暂停”按钮的图标。

（2）定义设置音量的方法：setVolume()。该方法中只要设置 Audio 对象的 volume 属性等于 range 控件的值即可。

【代码实现】

对项目的结构和样式有所了解后，即可开始编写代码来实现它。该项目的 HTML 页面代码如 code\06\0603\0603.html 所示。

0603.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>HTML5 音乐播放器</title>
    <link rel="stylesheet" href="css/audio.css" media="screen"
          type="text/css" />
     <link rel="stylesheet" href="css/font-awesome.min.css" media="screen"
           type="text/css" />
</head>
<body>
  <div id="container">
    <div id="player">
```

```
        <!--引入音频源文件-->
        <audio id="audio">
<source src="audio/陪你度过漫长岁月.mp3" type="audio/mpeg" codecs="mp3"/>
        </audio>
        <!--引入专辑封面图片 -->
        <img src="images/music.jpg">
        <!--制作显示歌词的按钮-->
        <label class="to-lyrics-label" for="to-lyrics">
        <i class="fa fa-caret-down fa-lg"></i></label>
        <input type="checkbox" id="to-lyrics"><!-- 歌词切换-->
        <div class="cover">
            <div class="controls">
  <button title="循环播放"> <i class="fa fa-retweet fa-lg"></i></button>
  <button title="上一曲"> <i class="fa fa-backward fa-2x"></i></button>
  <button id="play-pause" title="播放" onclick="togglePlayPause()">
   <i class="fa fa-play fa-3x"> </i></button><!--播放/暂停切换按钮-->
    <button title="下一曲"><i class="fa fa-forward fa-2x"></i></button>
    <button title="顺序播放"><i class="fa fa-random fa-lg"></i></button>
    <input name="volume" id="volume" min="0" max="1" step="0.1"
          type="range" onchange="setVolume()" /><!--音量控件-->
            </div><!-- #controls -->
            <div class="info">
    <p class="song"><a href="#" target="_blank">陪你度过漫长岁月</a></p>
    <p class="author"><a href="#" target="_blank">陈奕迅</a></p>
            </div><!-- #info -->
            <div class="lyrics">
                <p>陪你度过漫长岁月</p>
                <p>演唱：陈奕迅</p>

                <p>走过了人来人往 不喜欢也得欣赏</p>
                <p>我是沉默的存在 不当你世界 只做你肩膀</p>
                <p>拒绝成长到成长 变成想要的模样</p>
                <p>在举手投降以前 让我再陪你一段</p>

                <p>陪你把沿路感想活出了答案</p>
                <p>陪你把独自孤单变成了勇敢</p>
                <p>一次次失去又重来 我没离开</p>
                <p>陪伴是 最长情的告白</p>
                <p>陪你把想念的酸 拥抱成温暖</p>
                <p>陪你把彷徨 写出情节来</p>
                <p>未来多漫长 再漫长 还有期待</p>
                <p>陪伴你 一直到 故事给说完</p>

                <p>让我们静静分享 此刻难得的坦白</p>
                <p>只是无声地交谈 都感觉幸福 感觉不孤单</p>
            </div><!-- #lyrics -->
            <p class="scroll">收起</p>
        </div><!-- cover -->
    </div><!-- #player -->
</div><!-- #container -->
```

```
61   <script src="js/audio.js" charset="gbk"></script>
62 </body>
63 </html>
```

该项目的 CSS 样式代码如 code\06\0603\css\audio.css 所示。

audio.css

```
1  /*第 6 单元 项目 6-3 音乐播放器 css 文件*/
2  /*并排放置两个带边框的框时，可令边框和内边距包含在指定宽度和高度内，用来解决盒子被挤下去
     的问题*/
3  *, *:before, *:after {
4      box-sizing: border-box;
5  }
6  body {
7      margin: 0;
8      min-height: 100%;
9      background: #EEE;
10     font-family: 'Roboto Condensed', sans-serif;
11     font-weight: 300;
12 }
13 a { color: #FFF; text-decoration: none; }
14 a:hover { color: #26C5CB; }
15 p { margin: 0; }
16 #container,
17 #player,
18 .cover {
19       position: absolute;
20       margin: auto;
21       top: 0;
22       left: 0;
23       right: 0;
24       bottom: 0;
25 }
26 #container {
27       width: 320px;
28       height: 320px;
29 }
30 #player {
31      width: 300px;
32      height: 300px;
33      background: #fff;
34      border-radius: 50%;
35      overflow: hidden;
36      box-shadow: 2px 2px 20px 0 rgba(0,0,0,0.3);
37      z-index: 300;
38 }
39 /*专辑封面 */
40 img {
41     width: 100%;
42     height: 100%;
43     background: #fff;
```

```
    opacity: 0.75;/*透明度75%*/
}
/*“播放”按钮 */
.controls {
      position: relative;
      width: 100%;
      color: #fff;
      text-align: center;
}
button {
      margin: 5px;
      color: #fff;
      background: transparent;/*背景透明*/
      border: 0;
      outline: 0;
      cursor: pointer;
      text-align: center;
      text-shadow: 1px 1px 2px #000;
      /*过渡：所有属性都改变 在0.3s内完成 以慢速开始和结束*/
      transition: all 0.3s ease-in-out;
      -webkit-transition: all 0.3s ease-in-out;
}
button:hover {
      color: #26C5CB;
}
/* 歌曲信息*/
.info {
      position: relative;
      margin-top: 28px;
      bottom: 10px;
      color: #fff;
      text-align: center;
      text-shadow: 1px 1px 3px #000;
}
.song {
      font-size: 18px;
}
.author {
      font-size: 14px;
      margin-bottom: -8px;
}
/*以下3个属性写在一起表示该元素超出内容宽度后显示为省略号 */
.song, .author{
      white-space: nowrap;
      overflow: hidden;
      text-overflow: ellipsis;
}
/* Volume音量控件的样式 */
input[type='range'] {
      display: block;
```

```
94        margin: 14px auto;
95        width: 80px;
96       height: 2px;
97        outline: 0;
98        cursor: pointer;
99        box-shadow: 1px 1px 3px 0 #000;
100}
101/*伪元素::-webkit-slider-thumb 改变 range 的默认样式*/
102input[type='range']::-webkit-slider-thumb {
103     background: #AEAEAE;
104      width: 6px;
105     height: 6px;
106      border-radius: 50%;
107     transition: 0.1s all linear;
108      -webkit-transition: 0.1s all linear;
109     -webkit-appearance: none !important;
110}
111/*鼠标悬停时，该元素背景颜色变为#26C5CB，放大为原来的 2 倍*/
112input[type='range']:hover::-webkit-slider-thumb {
113     background: #26C5CB;
114     -webkit-transform:scale(2);
115}
116/*隐藏复选框*/
117input[type=checkbox] {
118       position: absolute;
119       top: -9999px;
120       left: -9999px;
121}
122label {
123       text-shadow: 1px 1px 3px #000;
124}
125.to-lyrics-label:hover {
126       color: #26C5CB;
127}
128label.to-lyrics-label {
129       position: absolute;
130       top: 276px;
131       left: 50%;
132       width: 20px;
133       height: 20px;
134       margin-left: -5px;
135       color: #fff;
136       cursor: pointer;
137       z-index: 500;
138}
139/* Lyrics */
140.lyrics {
141       position: relative;
142       width: 100%;
143       height: 96px;
```

```
    margin-top: 30px;
    padding: 4px 24px;
    color: #000;
    background: rgba(255,255,255,0.3);
    font-size: 12px;
    text-align: center;
    overflow-y: scroll;/*当内容超过 div 高度时，出现滚动条，内容滚动显示*/
    box-shadow: inset 0 -3px 5px 0 rgba(0,0,0,0.5);
    transition: all 0.5s ease-in-out;
    -webkit-transition: all 0.5s ease-in-out;
}
/*当鼠标悬停在歌词上时，背景变为白色 80%透明*/
.lyrics:hover {
   background: rgba(255,255,255,0.8);
}
/*清除滚动条样式*/
.lyrics::-webkit-scrollbar {
    display: none;
}
.scroll {
    color: #fff;
    text-align: center;
    font-size: 9px;
    font-weight: bold;
    text-shadow: 1px 1px 3px #000;
}
.cover {
    padding-top: 145px;
    transition: all 0.5s ease-in-out;
    -webkit-transition: all 0.5s ease-in-out;
}
/*~ 用于选取属性值中包含指定词汇的元素*/
#to-lyrics:checked ~ .cover {
    padding-top: 40px;
}
#to-lyrics:checked ~ .cover .lyrics {
    margin-top: 0px;
}
#to-lyrics:checked ~ .cover button {
    margin: 4px;
}
```

该项目的 JavaScript 代码如 code\06\0603\js\audio.js 所示。

audio.js

```
/*第 6 单元 项目 6-3 音乐播放器 js 文件*/
var audio = document.getElementById('audio');
var playpause = document.getElementById("play-pause");
var volume = document.getElementById("volume");
audio.controls = false;
//定义控制歌曲播放、暂停的方法：togglePlayPause()
```

```
//该方法中，在切换播放、暂停的同时，需要切换字体图标和按钮的 title 属性值
function togglePlayPause() {
    if (audio.paused || audio.ended) {
        playpause.title = "暂停";
        playpause.innerHTML = '<i class="fa fa-pause fa-3x"></i>';
        audio.play();
    } else {
        playpause.title = "播放";
        playpause.innerHTML = '<i class="fa fa-play fa-3x"></i>';
        audio.pause();
    }
}
//定义设置音量的方法：setVolume()，设置 Audio 对象的 volume 属性等于 range 控件的值
function setVolume() {
    audio.volume = volume.value;
}
```

【项目总结】

本项目的练习重点：

本项目主要为了巩固<audio>标签和 HTML DOM Audio 对象的应用，并且综合了前面学习的 CSS 内容，如字体图标、圆角边框、阴影、过渡等。

本项目的练习方法：

本项目只有“播放”按钮和调节音量的控件功能由 JavaScript 代码来实现，其他效果都是由 CSS 代码实现的。其实，JavaScript 也是可以操作 CSS 样式的，如 document.getElementByld ("to").style. backgroundColor="#fff"，有兴趣的读者可以动手尝试。

【思考题】

请简述 HTML5 中如何嵌入音频和视频，并列举 HTML5 支持的音频和视频格式。

Responsive Web Design

7 Chapter

单元 7
响应式 Web 设计

本书在前面的单元中讲解了如何使用 HTML5、CSS3 和 JavaScript 来制作网页。本单元将在 HTML5 和 CSS3 的知识的基础上，讲解一种新型网页设计理念——响应式 Web 设计。响应式网站可以针对不同的终端显示出合理的页面，实现一次开发、多处适用。之所以称之为新理念，是因为响应式不仅是一种跨终端的网页开发技术，还颠覆了之前的网页设计思想。本单元将针对响应式 Web 设计进行详细讲解。

【教学导航】

学习目标	1. 了解视口的概念 2. 掌握 CSS3 媒体查询的使用 3. 了解栅格系统 4. 掌握弹性盒布局
教学方式	读者一定要深刻理解前导知识中的内容。本单元项目代码非常多，读者要先熟悉网页结构，再进行样式添加。先添加基本样式，再针对响应式思想，用媒体查询进行调整。其中需要注意的是： 项目 7-1 中读者要掌握媒体查询的使用和汉堡菜单的实现； 项目 7-2 的重点是训练弹性盒布局的使用
重点知识	1. CSS3 媒体查询 2. 汉堡菜单的制作 3. 弹性盒布局
关键词	viewport、media、em、flex

【项目 7-1】 第一个响应式网站

【项目描述】

第一个响应式设计网站对于读者意义非凡，所以本项目选择了一个非常有意义的主题——环保网站。网页效果如图 7-1 所示。

图7-1 环保网站PC版

将浏览器窗口缩小到移动设备大小后，页面效果如图 7-2 所示。

图7-2　环保网站移动版

上面介绍了较有代表性的两版网页展示形态。实际上，该项目适用于多种屏幕大小，页面效果会随屏幕大小的改变而实时调整，这就是响应式 Web 设计。

【前导知识】

关于视口

视口在响应式设计中是一个非常重要的概念。在移动端浏览器中存在两种视口：一种是可见视口，即设备大小；另一种是视窗视口，即网页宽度。如图 7-3 所示，设备屏幕是 414px 的宽度。在浏览器中，414px 的屏幕宽度能够展示 1 200px 宽度的内容。那么，414px 就是可见视口的宽度，而 1 200px 就是视窗视口的宽度。

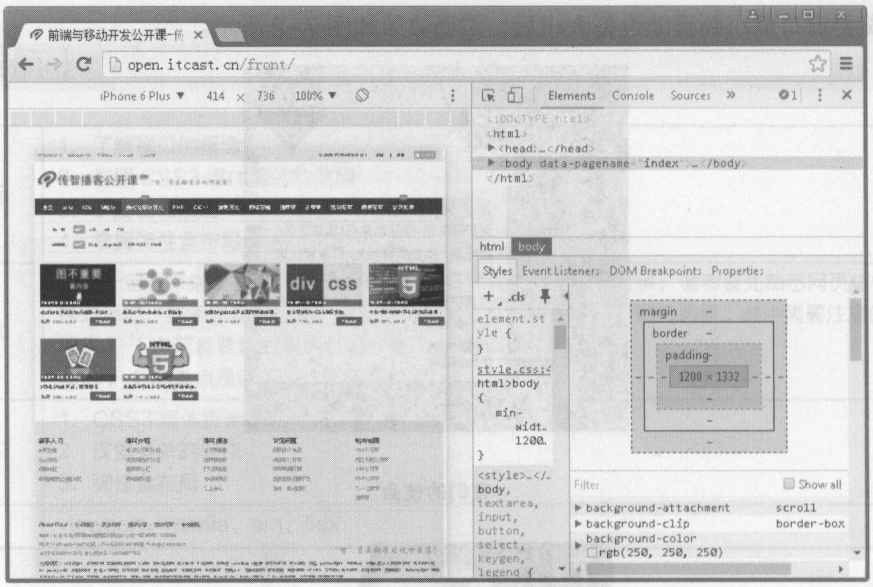


图7-3　视口

为了显示更多的内容，浏览器会经过 viewport 的默认缩放将网页等比例缩小。但是，为了让用户能够看清楚设备中的内容，通常情况下，并不使用默认的 viewport 进行展示，而是自定义配置视口的属性，使这个缩小比例更加适当。HTML5 中，<meta>标签可以用于配置视口属性。

```
<meta name="viewport" content="user-scalable=no, width=device-width,
      initial-scale=1.0, maximum-scale=1.0">
```

上述代码中，user-scalable 用于设置用户是否可以缩放，默认为 yes；width 用于设置视窗视口的宽度，device-width 表示视窗视口和可见视口宽度相同，该属性也可以设置成具体宽度；initial-scale 用于设置初始缩放比例，取值为 0 ~ 10.0；maximum-scale 用于设置最大缩放比例，取值为 0 ~ 10.0。除此之外，还可以通过 height 属性设置视窗视口的高度，minimum-scale 用于设置最小缩放比例。

媒体查询

之前，HTML4 和 CSS2 可以支持为不同的媒体类型（screen 屏幕和 print 打印）设置特定的 CSS 样式。在 CSS3 规范中，媒体查询可以根据视口宽度、设备方向等差异来改变页面的显示方式。媒体查询由媒体类型和条件表达式组成，示例代码如下所示。

```
@media screen and (max-width: 960px){
 /*样式设置*/
}
```

上述代码表示媒体类型为 screen 并且屏幕宽度小于等于 960px 时的样式。在实际开发中，通常会将媒体类型省略。接下来，我们通过一个案例演示一下媒体查询的具体用法，如 demo7-1 所示。

demo7-1.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
```

```
    <meta name="viewport" content="user-scalable=no,
          width=device-width, initial-scale=1.0, maximum-scale=1.0">
    <title>媒体查询</title>
    <style type="text/css">
       body {
          background-color: red;
          }
       @media (min-width: 320px){
          body {
                background-color: blue;
             }
       }
       @media (min-width: 414px){
          body {
                background-color: yellow;
                }
       }
       @media (min-width: 768px){
          body {
                background-color: grey;
                }
       }
       @media (min-width: 960px){
          body {
                background-color: pink;
                }
       }
    </style>
</head>
<body>
</body>
</html>
```

用浏览器打开 demo7-1，页面效果如图 7-4 ~ 图 7-7 所示。

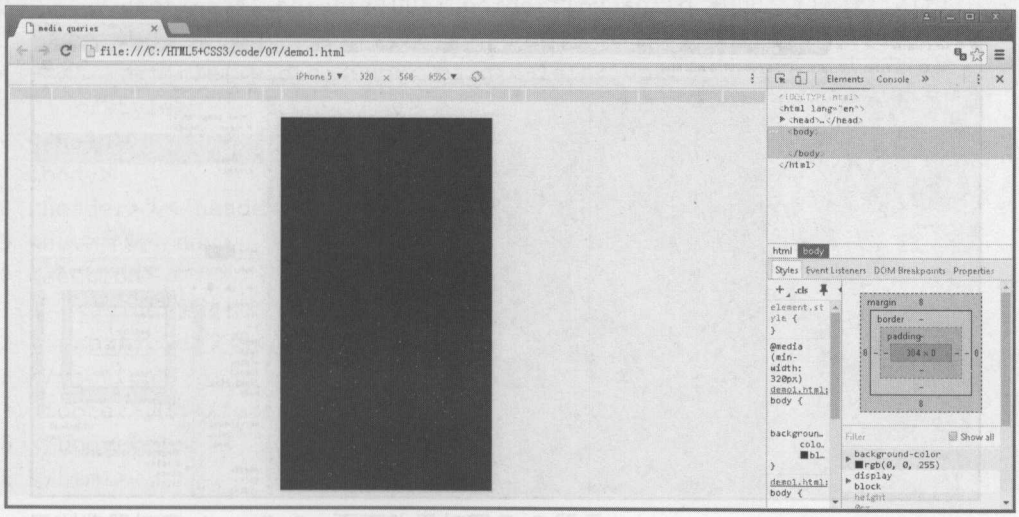


图7-4 iPhone5

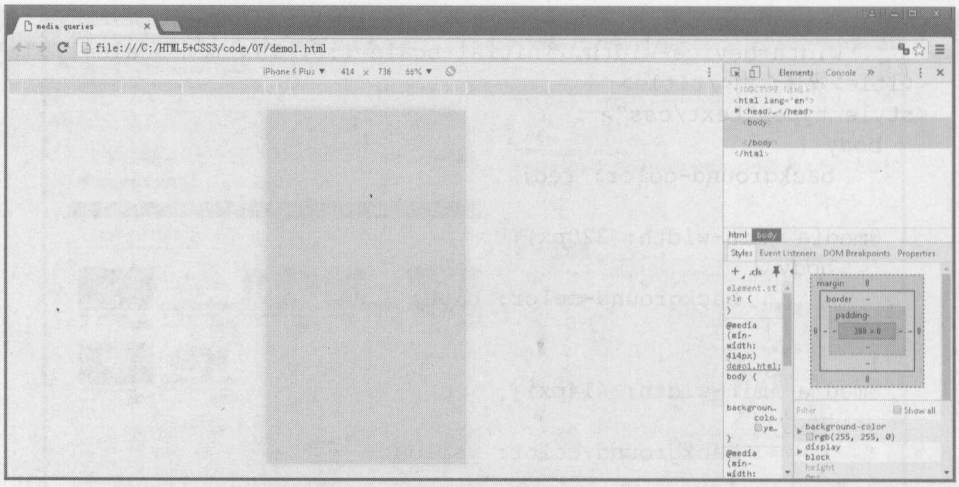


图7-5　iPhone6Plus

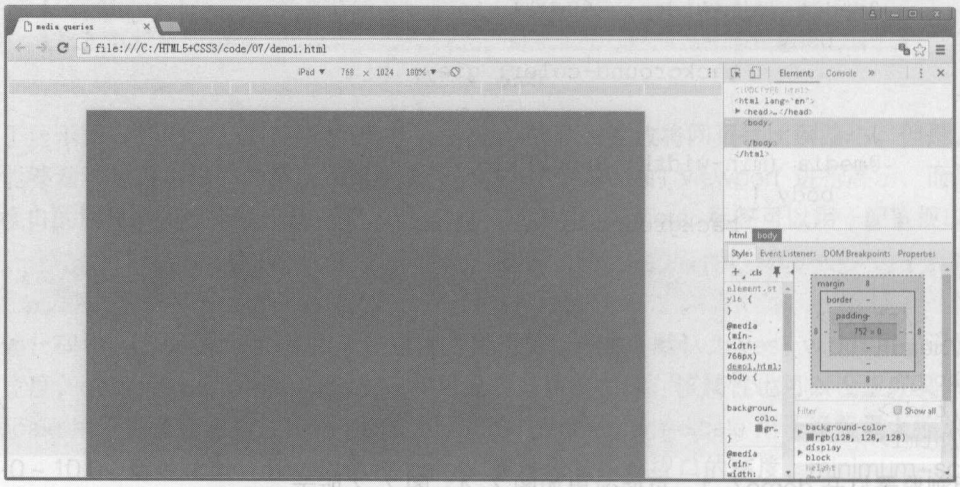


图7-6　iPad

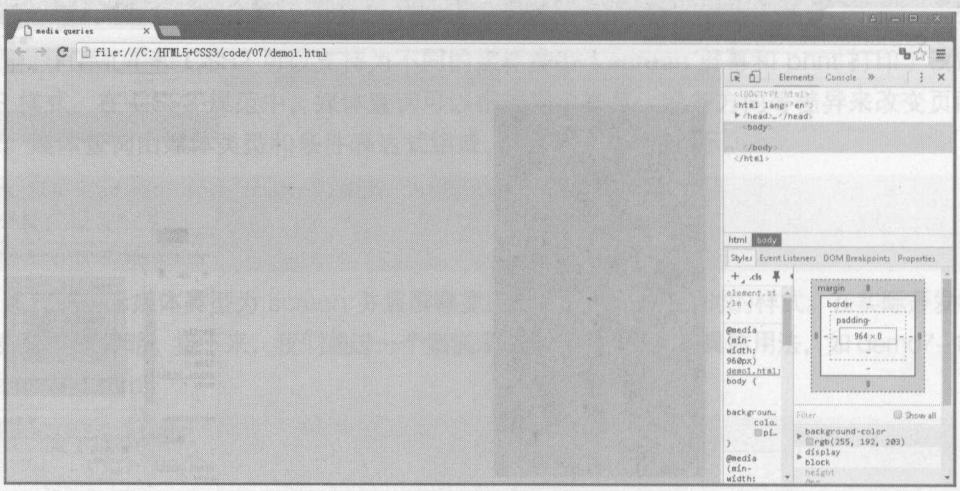


图7-7　PC页面

在 demo7-1 中，设定了当屏幕大于等于 320px 时，body 的背景色为蓝色；当屏幕大于等于 414px 时，body 的背景色为黄色；当屏幕大于等于 768px 时，body 的背景色为灰色；当屏幕大于等于 960px 时，body 的背景色为粉色。

需要注意的是，由于 CSS 代码是从上到下依次执行，所以当使用 min-width 来区分屏幕时，要按照从小屏到大屏的编写顺序；当使用 max-width 来区分屏幕时，要按照从大屏到小屏的编写顺序。

百分比布局

不要认为只使用媒体查询就能够制作出完美的响应式网站了，由于媒体查询只能针对某几个特定阶段的视口，在捕捉到下一个视口前，页面的布局是不会变化的，这样会影响页面的显示，同时也无法兼容日益增多的各种设备。所以，要想做出真正灵活的页面，还需要用百分比布局代替固定布局，并且使用媒体查询限制范围。

在此之前，我们习惯了可量取、可控制固定布局（以像素为单位）。然而，学习的过程就是知识迁移的过程，我们可以用将固定宽度换算为百分比宽度的方法，来帮助读者理解百分比布局。换算公式为：目标元素宽度 ÷ 父盒子宽度=百分数宽度。接下来，我们通过一个案例来演示将固定布局转换为百分比布局，如 demo7-2 所示。

demo7-2.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>固定布局转换为百分比布局</title>
    <style type="text/css">
        body>*{ width:980px; height:auto; margin:0 auto; margin-top:10px;
                border:1px solid #000; padding:5px;}
        header{ height:50px;}
        section{ height:300px;}
        footer{ height:30px;}
        section>*{ height:100%; border:1px solid #000; float:left;}
        aside{ width:250px;}
        article{ width:700px; margin-left:10px;}
    </style>
</head>
<body>
<header>头</header>
<nav>导航</nav>
<section>
    <aside>侧边栏</aside>
    <article>文章</article>
</section>
<footer>页脚</footer>
</body>
</html>
```

用浏览器打开 demo7-2，页面效果如图 7-8 所示。

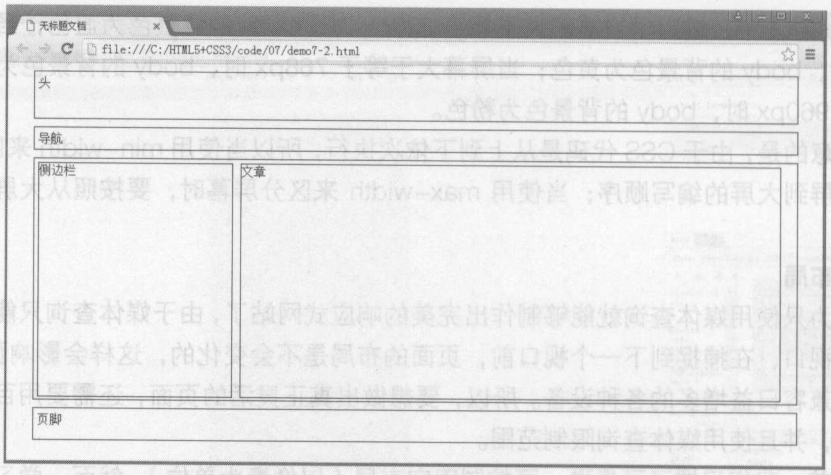


图7-8　固定布局

读者可以尝试改变浏览器窗口的大小，页面元素的大小是不会改变的。下面将 demo7-2 样式中的所有宽度修改为百分比的形式，代码如下所示。

```
<style type="text/css">
    body>*{ width:95%; height:auto; margin:0 auto; margin-top:10px;
     border:1px solid #000; padding:5px;}
    header{ height:50px;}
    section{ height:300px;}
    footer{ height:30px;}
    section>*{ height:100%; border:1px solid #000; float:left;}
    aside{ width:25.510204%;/*250÷980*/}
    article{ width:71.428571%; /*700÷980*/margin-left:10px;}
</style>
```

刷新页面，页面效果如图 7-9 所示。

在图 7-9 中，尝试缩小/放大浏览器，会发现网页随浏览器窗口的变化等比例缩小/放大。

与百分比布局算法类似的还有 em 单位，em 在之前的项目中已经使用过。用 em 代替 px 来设置字体大小有两点好处：一是能够缩放文字；二是 em 的实际大小是相对于父元素的字体大小而言的，如果在上文中设置字体为 100%，下文的字体都使用 em，则在页面整体发生改变时，下文中的字体都会随着上文的变化而变化。本项目就是使用 em 来设置各部分字体的大小。需要注意的是，用户的浏览器默认渲染的文字大小是 16px，所以在默认状态下，1em=16px。

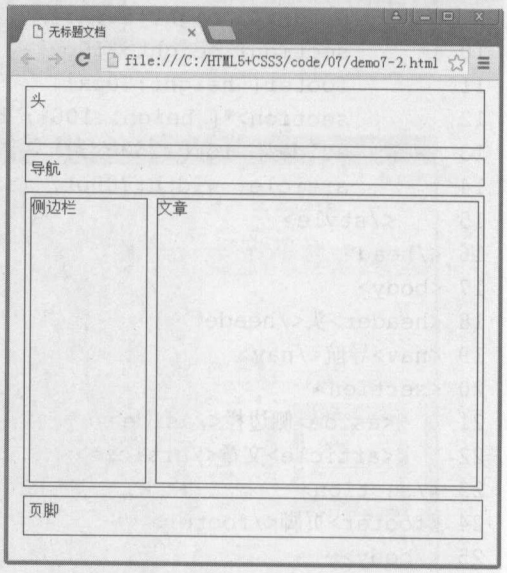


图7-9　百分比布局

【项目分析】

有了前导知识作为铺垫，接下来我们进行项目分析。该页面的页面标注和页面结构如图 7-10

和图 7-11 所示。

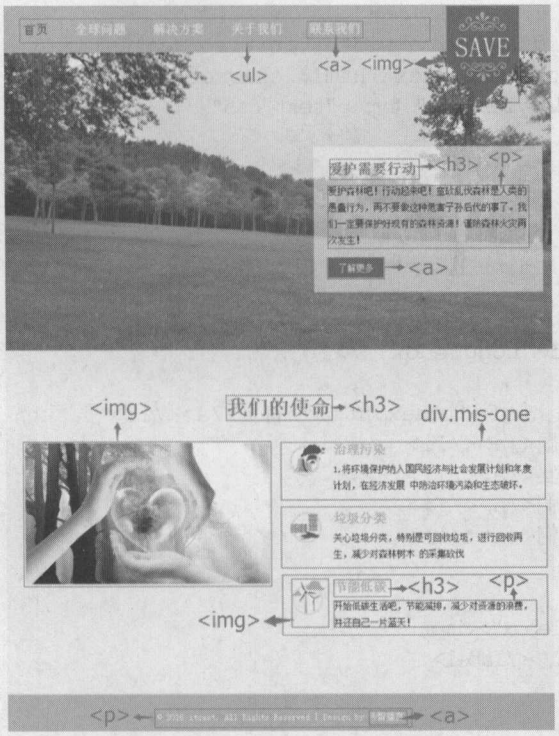


图7-10　页面标注

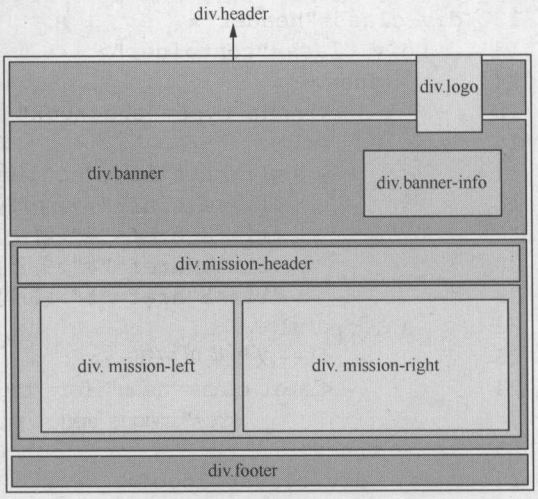


图7-11　页面结构

如图 7-11 所示的响应式页面由 header、banner、mission 和 footer 4 大部分构成。

该页面的实现细节具体分析如下。

（1）响应式页面各部分的宽度用百分比表示，如 header 的宽度我们设置为 100%。

（2）header 中包括导航菜单和 logo 左右两部分，其中 logo 部分使用绝对定位；在<nav>中嵌套<ul>列表制作导航菜单。

（3）当屏幕缩小到 640px 时，出现汉堡菜单按钮，该按钮使用<lable>标签嵌套<img>标签引入按钮图片，该按钮的功能使用 CSS 通过 checkbox 进行控制。

（4）banner 部分由 div.banner 嵌套 div.banner-info 构成，为 div.banner 设置背景图，当浏览器窗口缩小时，需要对 div.banner-info 设置媒体查询。

（5）在 PC 端 div.mission-left 和 div.mission-right 两部分横向排列，在移动端需要使用媒体查询将其纵向排列。

【代码实现】

对页面的结构有所了解后，即可开始编写代码来实现它。该项目的 HTML 代码如 code\07\0701\0701.html 所示。

0701.html

```
1  <!DOCTYPE html>
2  <html lang="en">
```

```
<head>
    <title>响应式绿色环保</title>
    <!-- 自定义主题文件 -->
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <link href="css/response.css" rel="stylesheet" type="text/css"
          media="all" />
</head>
<body>
<!-- header -->
<div class="header">
    <div class="container">
        <nav>
            <input type="checkbox" id="togglebox" />
            <ul>
                <li><a class="active" href="index.html">首页</a></li>
                <li><a href="#">全球问题</a></li>
                <li><a href="#">解决方案</a></li>
                <li><a href="#">关于我们</a></li>
                <li><a href="#">联系我们</a></li>
            </ul>
            <!--汉堡菜单按钮-->
            <label class="menu" for="togglebox"><img
                    src="images/menu. png"/></label>
        </nav>
        <div class="logo">
            <a href="index.html"><img src="images/logo.png"/></a>
        </div>
        <div class="clearfix"></div>
    </div>
</div>
<!-- //header -->
<!-- banner -->
<div class="banner">
    <div class="container">
        <div class="banner-info">
            <h3>爱护需要行动</h3>
            <p>爱护森林吧！行动起来吧！乱砍滥伐是人类的愚蠢行为，再也不要做这种危害子孙
                后代的事了。我们一定要保护好现有的森林资源！谨防森林火灾再次发生！</p>
            <a href="#" class="button">了解更多</a>
        </div>
    </div>
</div>
<!-- //banner -->
<!-- mission -->
<div class="mission">
    <div class="container">
        <div class="mission-header">
            <h3>我们的使命</h3>
        </div>
        <div class="mission-container">
            <div class="mission_div mission-left">
```

```
52              <img src="images/mission_img.jpg" alt=""/>
53          </div>
54          <div class="mission_div mission-right">
55              <div class="mis-one">
56                  <div class="mis-left">
57                      <img src="images/i1.gif" alt=""/>
58                  </div>
59                  <div class="mis-right">
60                      <h3>治理污染</h3>
61                      <p>1.将环境保护纳入国民经济与社会发展计划和年度计划，在经济发展
62                          中防治环境污染和生态破坏。 </p>
63                  </div>
64                  <div class="clearfix"></div>
65              </div>
66              <div class="mis-one">
67                  <div class="mis-left">
68                      <img src="images/i2.gif" alt=""/>
69                  </div>
70                  <div class="mis-right">
71                      <h3>垃圾分类</h3>
72                      <p>关心垃圾分类，特别是可回收垃圾，进行回收再生，减少对森林树木
73                          的采集砍伐</p>
74                  </div>
75                  <div class="clearfix"></div>
76              </div>
77              <div class="mis-one">
78                  <div class="mis-left">
79                      <img src="images/i3.gif" alt=""/>
80                  </div>
81                  <div class="mis-right">
82                      <h3>节能低碳</h3>
83                  <p>开始低碳生活吧！节能减排，减少对资源的浪费，并还自己一片蓝天！</p>
84                  </div>
85                  <div class="clearfix"></div>
86              </div>
87          </div>
88          <div class="clearfix"></div>
89      </div>
90  </div>
91 </div>
92 <!-- //mission -->
93 <!-- footer -->
94 <div class="footer">
95     <div class="container">
96         <p> &copy; 2016 itcast. All Rights Reserved | Design by<a href="#">
               传智播客</a></p>
97         <div class="clearfix"></div>
98     </div>
99 </div>
100</body>
101</html>
```

该项目的 CSS 代码如 code\07\0701\css\response.css 所示。

response.css

```
/*第 7 单元 项目 7-1 响应式绿色环保 CSS 文件*/
html {
    font-family: sans-serif;
    /*为元素指定的任何内边距和边框都将在已设定的宽度和高度内进行绘制*/
    box-sizing: border-box;
}
*, *:before, *:after {
    /*规定应从父元素继承 box-sizing 属性的值*/
    box-sizing: inherit;
}
body,p,ul,li,dl,dt,dd,h1,h2,h3,h4,img{
    margin:0;
    padding:0;
    border:0;
}
ul,li{
    list-style-type:none;
}
body{
    width:100%;
    background:#fff;
    font-family: 'Roboto Slab', serif;
}
body a{
    text-decoration:none;
    transition:0.5s all;/*过渡时长为 0.5s*/
}
img {
    max-width:100%;/*图片的最大宽度为 100%*/
    height: auto;
    width: auto;
    vertical-align: middle;/*把此元素放置在父元素的中部*/
}
.clearfix{
    clear: both;/*清除浮动*/
}
.container{
    margin: 0 auto;
    padding: 0 15px;
    width:100%;
}
/*--header、nav--*/
.header{
    width:100%;
    background:#7ddf6c;
    padding:33px 0;
    position:relative;
```

```
}
nav ul {
    margin: 0;
    padding: 0;
}
nav ul li{
    margin:0 35px;
    display:inline-block;/*将其设置为块级元素*/
}
nav ul li a{
    font-family: 'Droid Serif', serif;
    color: #fff;
    font-size: 1.25em;/*20px÷16px=1.25*/
    font-weight:500;/*定义字体粗细*/
}
nav ul li a:hover,nav ul li a.active{
    color:#000;
}
/* 复选框用于切换菜单的开合状态 */
nav input[type="checkbox"] ,
.menu{
    position: absolute;/*相对于父元素绝对定位*/
    left: 2%;
    top: 10px;
    display:none;/*隐藏不显示*/
}
.logo{
    position:absolute;/*绝对定位，设置宽高*/
    right: 10%;
    top: 0%;
}
/*--banner--*/
.banner{
    width:100%;
    background:url(../images/banner.jpg) no-repeat 0px 0px;
    background-size:cover;
    min-height: 680px;
    overflow: hidden;
}
.banner-info{
    width: 30%;
    background:rgba(255, 255, 255, 0.65);
    padding: 30px 30px;
    float:right;
    margin-top: 320px;
}
.banner-info h3{
    font-family: 'Droid Serif', serif;
    font-size: 1.5em; /*24px÷16px=1.5*/
    color: #159400;
```

```
}
.banner-info p{
    font-size:0.875em; /*14px÷16px=0.875*/
    line-height:1.8em;
    color:#000;
    margin: 9px 0 15px;
}
.banner-info a{
    display: inline-block;/*将其设置为块级元素*/
    padding:7px 15px;
    background: #159400;
    font-size:1em;
    color:#ffffff;
}
a.button:hover,
a.button:focus,
a.button:active {
    background: #6cd79c;
    text-decoration: underline;
}
/*--mission--*/
.mission {
    background:#FFFDD2;
    padding: 80px 0;
}
.mission-header h3 {
    font-family: 'Droid Serif', serif;
    font-size: 2em;
    color: #159400;
    text-align: center;
}
.mission-container{
    margin-top: 35px;
}
.mission_div{
    width: 50%;
    float: left;
    position: relative;
    min-height: 1px;
    padding:0 15px;
}
.mission-left img {
    width: 100%;
}
.mis-one {
    margin-bottom: 30px;
}
.mis-left {
    width: 15%;
    float: left;
```

```
}
.mis-left img {
    width: 100%;
}
.mis-right {
    width: 82%;
    float: right;
}
.mis-right h3 {
    font-size: 1.25em; /*20px÷16px=1.25*/
    color: #7DDF6C;
}
.mis-right p {
    font-size: 0.875em;/*14px÷16px=0.875*/
    color: #000;
    line-height: 1.8em;
    margin: 12px 0 0 0;
}
/*--footer--*/
.footer {
    padding: 18px 0;
    background: #7ddf6c;
}
.footer p {
    margin: 9px 0 0 0;
    font-size: 0.875em;
    color: #fff;
    text-align: center;
}
.footer p a {
    color: #fff;
}
.footer p a:hover,.footer p a.active{
    color:#000;
}
/*-- responsive media queries --*/
@media (max-width: 1440px){
    .banner-info {
        width: 34%;
        padding: 22px;
        margin-top: 280px;
    }
}
@media (max-width: 1280px) {
    .banner-info {
        width: 36%;
        padding: 22px;
        margin-top: 200px;
    }
}
```

```
@media (max-width: 1200px){/*当屏幕小于 1200px 时*/
    .header {
        padding: 24px 0;
    }
    .logo {
        right: 6%;
        width: 13%;
    }
    nav > ul li {
        margin: 0 17px;
    }
    .logo img {
        width: 100%;
    }
    .banner {
        min-height: 456px;
    }
    .mission{
        padding: 70px 0;
    }
    .mis-right p {
        margin: 6px 0 0 0;
    }
    .mis-one {
        margin-bottom: 26px;
    }
}
@media (max-width: 1024px){
    .banner-info {
        padding: 22px 22px;
        margin-top: 145px;
        width: 43%;
    }
    .banner-info h3 {
        font-size: 1.4em
    }
    .banner-info a {
        font-size: 0.875em;
    }
}
@media (max-width: 768px){/*当屏幕小于 768px 时*/
    nav > ul li a {
        font-size: 1em;
    }
    .banner {
        min-height: 360px;/*限制 banner 的最小高度为 360px*/
    }
    .banner-info {
        width: 55%;
        padding: 13px;
```

```
        margin-top: 88px;
    }
    .banner-info h3 {
        font-size: 1.25em
    }
    .banner-info p{
        font-size: 0.75em
    }
    .banner-info a {
        font-size: 0.875em;
    }
    .mission{
        padding: 55px 0;
    }
    .mission-left,.mission-right {
        float: left;
        width: 50%;
    }
    .mission-right {
        padding-left: 0;
    }
    .mis-right {
        width: 80%;
        float: right;
    }
    .mis-left {
        width: 16%;
    }
    .mis-one {
        margin-bottom: 20px;
    }
}
@media (max-width: 640px){/*当屏幕小于 480px 时*/
    .header {
        padding: 25px 0;
    }
    .menu {
        display: block;/*显示该对象，之前被隐藏*/
        cursor: pointer;/*设定鼠标指针的形状为一只伸出食指的手*/
    }
    .menu img{
        width:100%;
    }
    nav > ul {
        display: none;
    }
    nav input[type="checkbox"]:checked ~ ul {
        display: block;
    }
    nav ul li {
```

```
298         width: 100%;
299         display: inline-block;
300         text-align: center;
301         margin: 0;
302         padding:0;
303     }
304     nav ul li a {
305         display:block;
306         margin: 10px 0;
307     }
308     .logo {
309         width: 17%;
310         right: 4%;
311     }
312     .banner {
313         min-height: 300px;
314     }
315     .banner-info h3 {
316         font-size: 1em
317     }
318     .mission {
319         padding: 45px 0;
320     }
321     .mission-header h3{
322         font-size: 1.25em;
323     }
324     .mis-right h3 {
325         font-size: 1em;
326     }
327     .mission-left, .mission-right {
328         padding: 0;
329         float: left;
330         width: 100%;
331     }
332     .mission-right {
333         margin: 30px 0 0 0;
334     }
335     .mis-left {
336         width: 13%;
337         margin-top: 3px;
338     }
339     .mis-right {
340         width: 82%;
341     }
342     .mis-right p {
343         margin: 10px 0 0 0;
344     }
345     .mis-one {
346         margin-bottom: 24px;
347     }
```

```
348    .mis-one:nth-child(3) {
349        margin: 0;
350    }
351    .footer p {
352        margin: 0px 0 20px 0;
353    }
354}
```

【项目总结】

本项目的练习重点：

通过本项目的练习，读者应该学会使用百分比布局、媒体查询、视口属性的设置等来调整响应式网站的变化。本项目要求读者理解响应式设计的各个技术点，打好响应式设计基础。后面的单元中会使用 bootstrap 框架轻松、便捷地实现响应式 Web 设计。

本项目的练习方法：

本项目代码量虽然很多，但实现起来并不复杂。建议读者先完成已有知识能解决的问题（页面的框架和布局），再通过媒体查询来调整浏览器窗口变小导致的模块样式冲突等问题。

本项目的注意事项：

- 制作响应式网站一定要添加视口设置，所以读者要先明确视口设置的每一个属性代表的意义。
- 汉堡菜单在移动端设计的出现率非常高，读者要了解其实现，并可以自行添加一些动画。
- 媒体查询对于网页的调整，最需要的是代码细致并保证页面有平缓的变化。

【项目 7-2】 社交网站个人信息页面

【项目描述】

由于生活节奏越来越快，人们经常通过社交网站结交朋友、组织一些聚会等。本项目将带领读者实现一个响应式社交网站的个人信息页面，如图 7-12 所示。

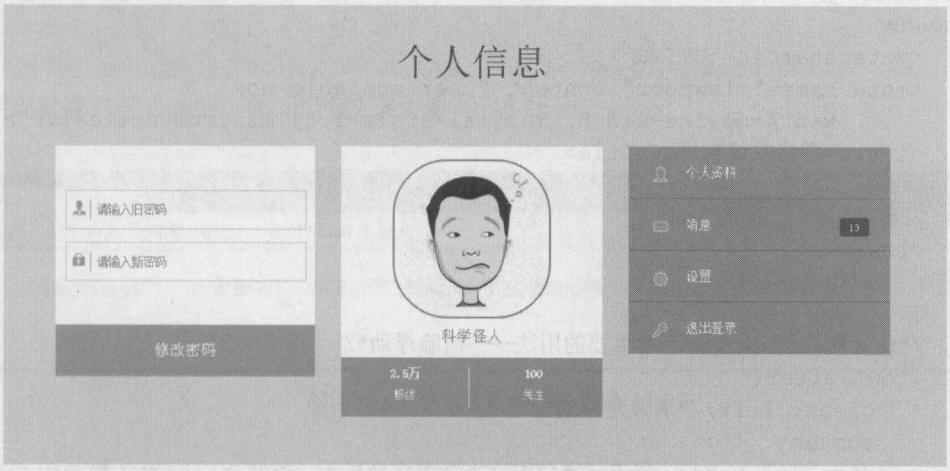


图7-12　个人信息页面-PC版

将浏览器窗口缩小到移动设备大小，页面效果如图 7-13 所示。

图7-13　个人信息页面-移动版

上面展示了 PC 和移动两版网页展示形态，实际上该项目适用于多种屏幕大小，页面效果会随屏幕大小的改变而实时调整。

【前导知识】

响应式栅格系统

在网页制作中，栅格系统（又称网格系统）就是用固定的格子进行网页布局，是一种清晰、工整的设计风格。栅格系统最早应用于印刷媒体上，后来被应用于网页布局中，而随着响应式设计的流行，栅格系统开始被赋予新的意义，即一种响应式设计的实现方式。下面，我们通过图 7-14，来给读者以响应式栅格系统的第一印象。

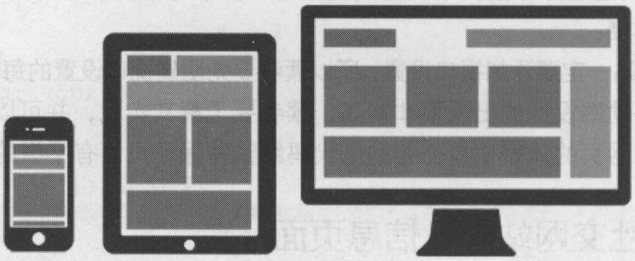


图7-14　响应式栅格系统

接下来，我们通过一个案例来演示栅格系统在响应式设计中的应用，如 demo7-3 所示。

demo7-3.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="user-scalable=no,
          width=device-width, initial-scale=1.0, maximum-scale=1.0">
    <title>栅格系统布局</title>
</head>
<style type="text/css">
    .row{
        width: 100%;
    }
    /*伪元素:after 的一个很重要的用法——清除浮动*/
    .row :after{
        clear: left;/*清除左浮动*/
        content: '';
        display: table;/*该元素会作为块级表格来显示（类似 <table>）*/
    }
```

```
    /*CSS3 新增[attribute^=value] 选择器，用于匹配属性值以指定值开头的每个元素。*/
    [class^="col"]{
        float: left;
        background-color: #e0e0e0;
    }
    .col1{
        width: 25%;
    }
    .col2{
        width: 50%;
    }
    @media (max-width: 768px) {
        .row{
            width: 100%;
        }
        [class^="col"]{
            float: none;
            width:100%;
        }
    }
</style>
<body>
<div class="row">
    <header>页头</header>
</div>
<div class="row">
    <nav class="col1">导航</nav>
    <div class="col2">主要内容</div>
    <aside class="col1">侧边栏</aside>
</div>
<div class="row">
    <footer>页尾</footer>
</div>
</body>
</html>
```

用浏览器打开 demo7-3，页面效果如图 7-15 所示。

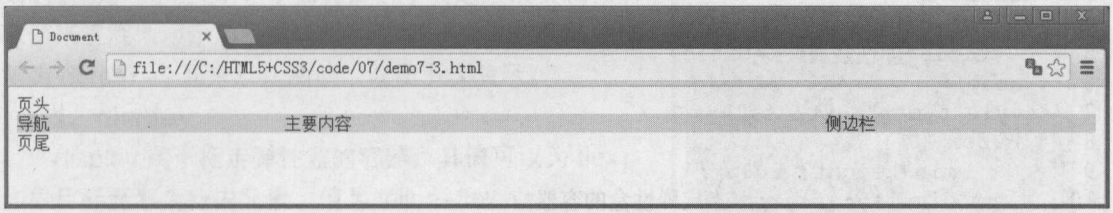


图7-15 栅格系统布局PC端页面效果

将浏览器窗口缩小至 375px 时，页面效果如图 7-16 所示。

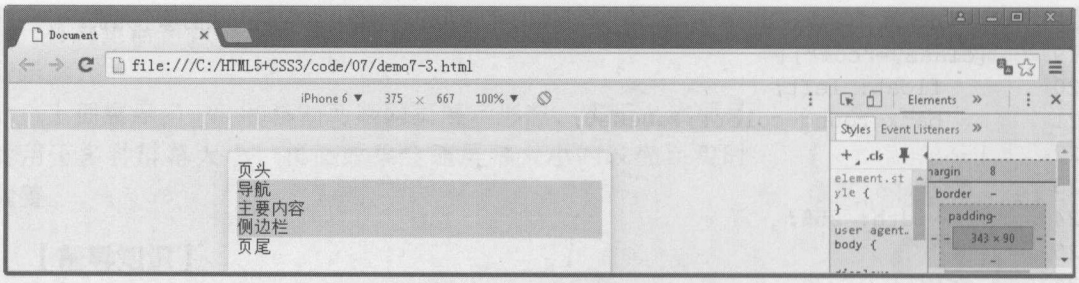


图7-16　栅格系统布局移动端页面效果

栅格系统作为一种设计理念，在第 8 单元将要介绍的 Bootstrap 中得到了很好的应用，这里读者了解栅格系统的布局概念即可。

弹性盒布局

说到响应式，就不得不提 CSS3 中的弹性盒布局。通过它可以轻松地创建响应式网页布局，为盒状模型增加灵活性。弹性盒改进了块模型，既不使用浮动，也不会在弹性盒容器与其内容之间合并外边距，是一种非常灵活的布局方法。首先，我们看一下弹性盒的结构，如图 7-17 所示。

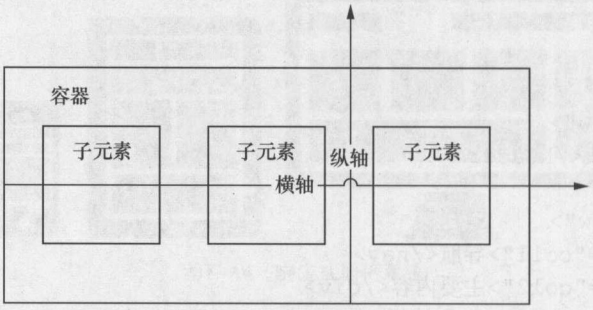


图7-17　弹性盒结构

从图 7-17 中可以看出，弹性盒由容器、子元素和轴构成，并且默认情况下，子元素的排布方向与横轴的方向是一致的。

接下来，我们通过一个案例来演示弹性盒控制布局的各个属性的应用，如 demo7-4 所示。

demo7-4.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>弹性盒属性</title>
</head>
<style type="text/css">
    .box {
        min-height: 200px;
        display: flex;/*指定弹性盒的容器*/
        /*
        flex-direction: row|row-reverse|column|column-reverse;
        flex-wrap: nowrap|wrap|wrap-reverse;
         */
        flex-flow: row;/*弹性盒子元素按横轴方向顺序排列*/
```

```
        justify-content: center;/*设置弹性盒子元素向行中间位置对齐*/
        align-items: center;/*弹性盒子元素向垂直于轴的方向上的中间位置对齐*/
        background-color: gray;
    }
    .A,.B,.C {
        background-color: white;
        border:1px solid gray;
    }
    .box div.A {
        order: 1;/*order 设置该子元素出现的顺序*/
        flex-grow: 0;/*扩展比率*/
        flex-shrink: 1;/*收缩比率*/
        flex-basis: auto;/*宽度，像素值*/
    }
    .box div.B {
        order: 2;
        flex: 0 1 auto;/*扩展比率 0、收缩比率 1 和宽度居中的缩写形式*/
    }
    .box div.C {
        order: 3;
        flex: 0 1 auto;
    }
</style>
<body>
<div class="box">
    <div class="A">A</div>
    <div class="B">B</div>
    <div class="C">C</div>
</div>
</body>
</html>
```

用浏览器打开 demo7-4，页面效果如图 7-18 所示。

从 demo7-4 中可以看出，弹性盒提供了很多属性来设置子元素显示的方向、顺序、宽高比例等。下面，我们就针对 demo 中样式里的属性进行详细的讲解。首先声明，下面的讲解中会通过改变属性的取值来直观地演示属性的特性，在每一次改变后都会恢复为 demo7-4 原来默认的值，以便下次演示。

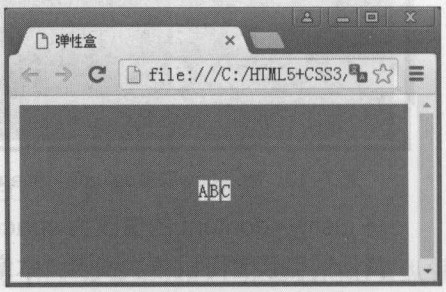


图7-18　弹性盒属性

1. display

display 用于指定弹性盒的容器，其值可以为 flex；如果目标元素为行内元素，值为 inline-flex。

2. flex-flow

flex-flow 是属性 flex-direction 和 flex-wrap 的简写，用于排列弹性子元素。其取值分别如表 7-1 和表 7-2 所示。

表 7-1　flex-direction 的取值

取值	描　述
row	弹性盒子元素按横轴方向顺序排列，默认值
row-reverse	弹性盒子元素按横轴方向逆序排列
column	弹性盒子元素按纵轴方向顺序排列
column-reverse	弹性盒子元素按纵轴方向逆序排列

表 7-2　flex-wrap 的取值

取值	描　述
nowrap	flex 容器为单行，该情况下 flex 子项可能会溢出容器
wrap	flex 容器为多行，flex 子项溢出的部分会被放置到新行
wrap-reverse	反转 wrap 排列

例如，将 flex-flow 的值改为 column-reverse，刷新浏览器，如图 7-19 所示。

3. justify-content

justify-content 属性能够用来设置子元素在当前轴方向的排列，其取值如表 7-3 所示。

表 7-3　justify-content 的取值

取值	描　述
flex-start	弹性盒子元素将向行起始位置对齐
flex-end	弹性盒子元素将向行结束位置对齐
center	弹性盒子元素将向行中间位置对齐
space-between	弹性盒子元素会平均地分布在行里，第一个元素里的边界与行的起始位置边界对齐，最后一个元素的边界与行结束位置的边距对齐
space-around	弹性盒子元素会平均地分布在行里，两端保留了元素与子元素之间间距大小的一半

例如，将 justify-content 的值改为 flex-start，刷新浏览器，如图 7-20 所示。

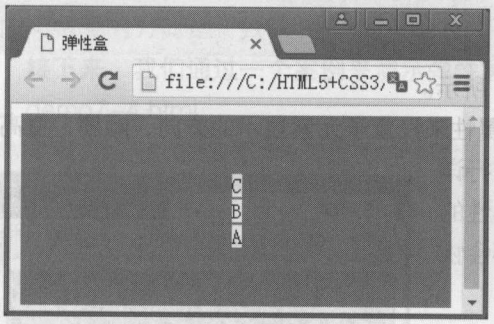


图7-19　flex-flow取值column-reverse

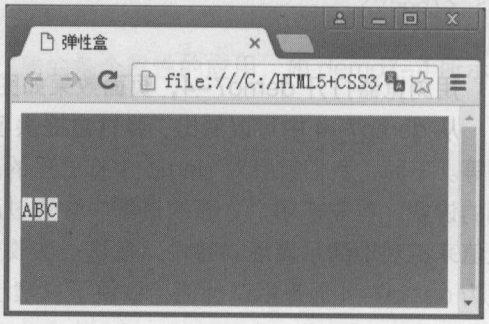


图7-20　justify-content取值flex-start

将 justify-content 的值改为 space-between，刷新浏览器，如图 7-21 所示。

其他的值读者可以依次试验，本章节不再赘述。

4. align-items

align-items 属性用于设置子元素在垂直于轴的方向上的排列，其取值如表 7-4 所示。

例如，将 align-items 的值设置为 flex-end，如图 7-22 所示。

表 7-4 align-items 的取值

取值	描 述
flex-start	弹性盒子元素向垂直于轴的方向上的起始位置对齐
flex-end	弹性盒子元素向垂直于轴的方向上的结束位置对齐
center	弹性盒子元素向垂直于轴的方向上的中间位置对齐
baseline	如果弹性盒子元素的行内轴与侧轴为同一条，则该值与'flex-start'等效。其他情况下，该值将参与基线对齐
stretch	如果指定侧轴大小的属性值为“auto”，则其值会使项目的边距盒的尺寸尽可能接近所在行的尺寸，但同时会遵照“min/max-width/height”属性的限制

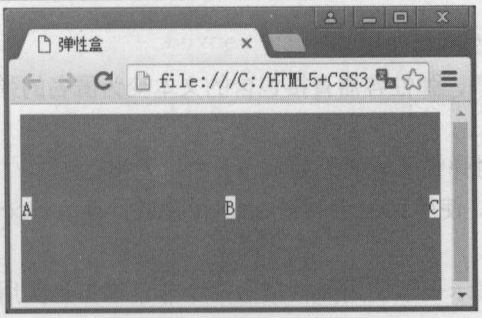


图7-21 justify-content取值space-between

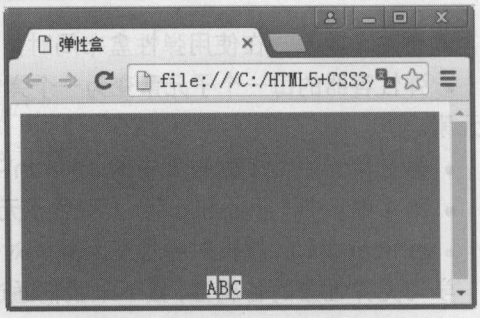


图7-22 align-items取值flex-end

5. order

order 属性用于设置子元素出现的顺序。例如，将 A、B、C 的 order 值分别改为 2、3、1，如图 7-23 所示。

6. flex

flex 属性是 flex-grow（扩展比率）、flex-shrink（收缩比率）和 flex-basis（宽度，像素值）的缩写，能够用来设置子元素的伸缩性。例如，将 A 的 flex-grow 改为 2，如图 7-24 所示。

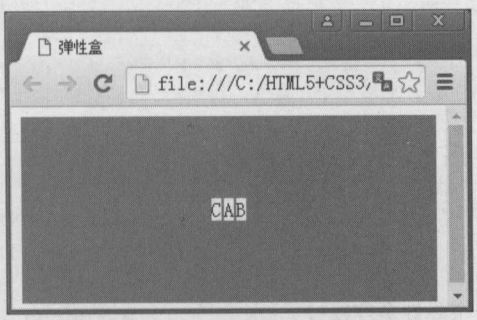


图7-23 将A、B、C的order值分别改为2、3、1

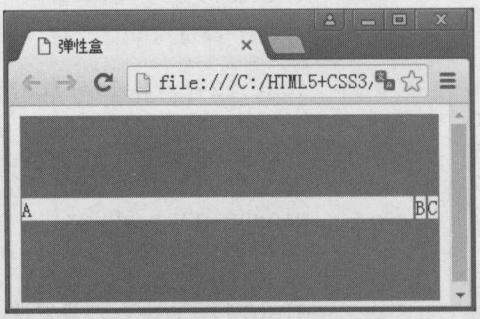


图7-24 将A的flex-grow改为2

将 A 的 flex-grow 值还原，将 A 的 flex-basis 改为 30px，如图 7-25 所示。

7. align-self

align-self 属性能够覆盖容器中的 align-items 属性，用于设置单独的子元素如何沿着纵轴排列。其取值有 auto、flex-start、flex-end、center、baseline、stretch，每个值的意义与 align-items 属性的取值类似。例如，将 A 和 C 的 align-self 设置为 center，B 的 align-self 设置为 stretch，如图 7-26 所示。

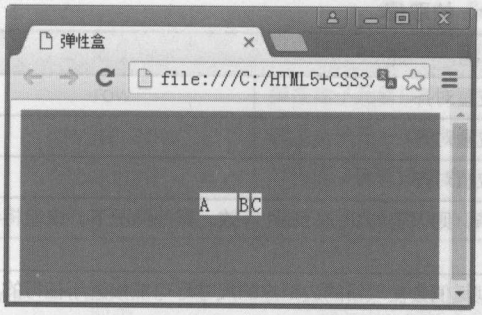


图7-25　将A的flex-basis改为30px

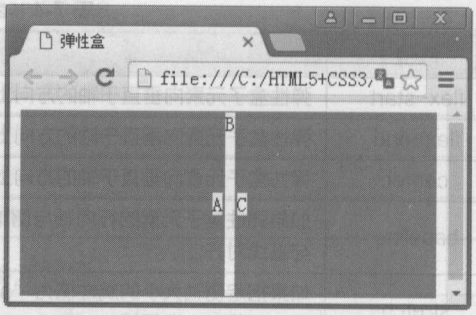


图7-26　align-self属性效果演示

需要注意的是，在使用弹性盒布局时，以下属性不起作用。

- 弹性容器的每一个子元素变为一个弹性子元素，弹性容器直接包含的文本变为匿名的弹性子元素。
- 第 2 单元中，多列布局中的 column-*属性对弹性子元素无效。
- 第 1 单元中，float 和 clear 对弹性子元素无效。使用 float 会导致 display 属性计算为 block。
- vertical-align 属性对弹性子元素的对齐无效。

学习完弹性盒的各属性，接下来我们通过案例演示一下使用弹性盒做一个非常常见且实用的响应式布局，如 demo7-5 所示。

demo7-5.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="user-scalable=no, width=device-width,
          initial-scale=1.0, maximum-scale=1.0">
    <title>弹性盒布局</title>
    <style>
        body {
            font: 24px Helvetica;
            background: #fff;
        }
        .main {
            min-height: 500px;
            margin: 0px;
            padding: 0px;
            display: flex;/*设置该 div 为一个弹性盒容器*/
            flex-flow: row;/*子元素按横轴方向顺序排列*/
        }
        .main > article {
            margin: 4px;
            padding: 5px;
            border-radius: 7pt;/*pt 也是文字大小的一种单位，1pt=px*3/4 */
            background: #719DCA;
            flex: 3 ;/*用数字也可以达到分配宽度的效果，将容器分为 5 份，占 3 份*/
            order: 2;/*排序为第 2 个子元素*/
```

```
        }
        .main > nav {
            margin: 4px;
            padding: 5px;
            border-radius: 7pt;
            background: #FFBA41;
            flex: 1 ;/*将容器分为 5 份，占 1 份*/
            order: 1;/*排序为第 1 个子元素*/
        }
        .main > aside {
            margin: 4px;
            padding: 5px;
            border-radius: 7pt;
            background: #FFBA41;
            flex: 1 ;/*将容器分为 5 份，占 1 份*/
            order: 3;/*排序为第 3 个子元素*/
        }
        header, footer {
            display: block;
            margin: 4px;
            padding: 5px;
            min-height: 100px;
            border: 2px solid #FFBA41;
            border-radius: 7pt;
            background: #FFF;
        }
        @media all and (max-width: 640px) {/*当屏幕小于 640px 时*/
            .main {
                flex-flow: column; /*弹性盒中的子元素按纵轴方向排列*/
            }
            .main > article, .main > nav, .main > aside {
                order: 0; /*将子元素都设置成同一个值，指按自然顺序排列*/
            }
            .main > nav, .main > aside, header, footer {
                min-height: 50px;
                max-height: 50px;
            }
        }
    </style>
</head>
<body>
    <header>header</header>
    <div class="main">
        <article>article</article>
        <nav>nav</nav>
        <aside>aside</aside>
    </div>
    <footer>footer</footer>
</body>
</html>
```

用浏览器打开 demo7-5，页面效果如图 7-27 所示。

将浏览器窗口缩小至 640px 以内，页面效果如图 7-28 所示。

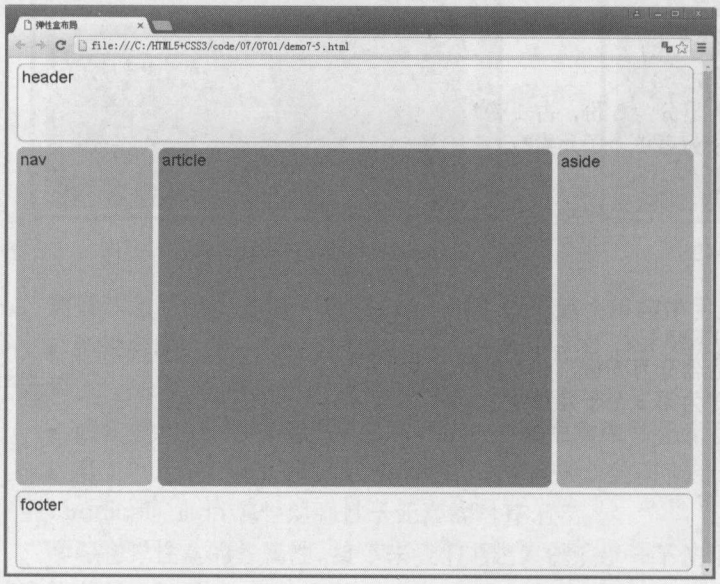


图7-27　弹性盒布局PC端页面效果

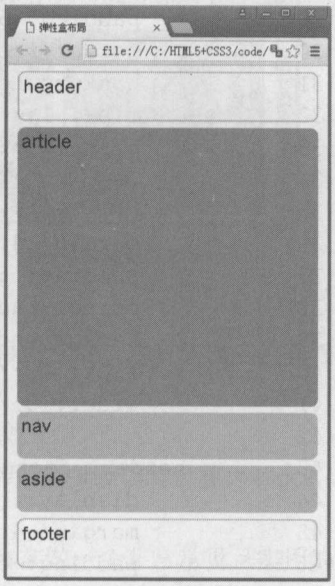


图7-28　弹性盒布局移动端页面效果

【项目分析】

该页面主要练习弹性盒布局知识，页面标注和页面结构如图 7-29 和图 7-30 所示。

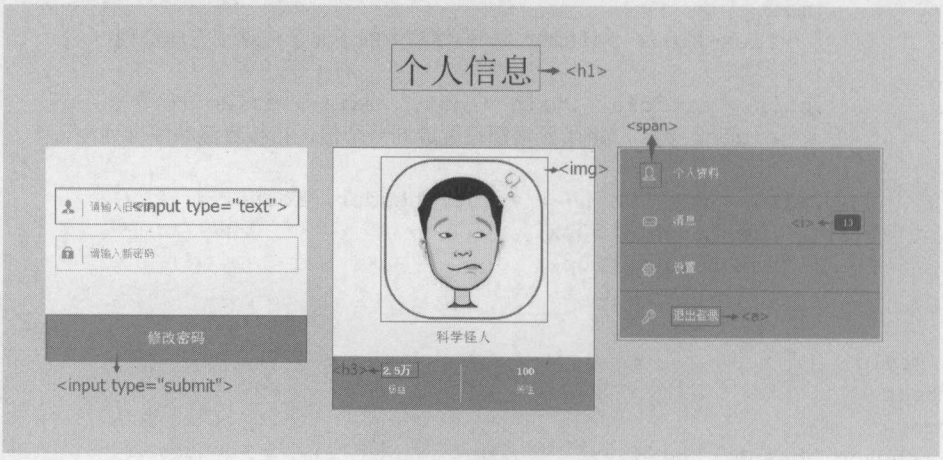


图7-29　页面标注

如图 7-30 所示的社交网站的个人信息页面主要由 div.element 嵌套 3 个 标签构成页面的 3 个模块。

该页面的实现细节具体分析如下。

（1）使用弹性盒对页面进行布局，将 div.element 设置为弹性盒容器，将 div. element-left、div.element-right 和 div.element-last 作为该容器的 3 个子元素。

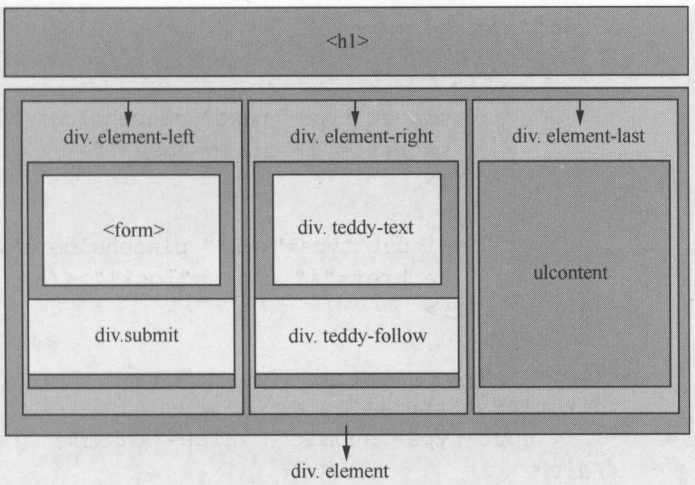


图7-30　页面结构

（2）在 PC 端子元素按横轴方向顺序排列，并且使用媒体查询，在浏览器窗口小于或等于 640px 时，子元素按纵轴方向排列。

（3）在 div.element-left 中添加表单和“提交”按钮，并且使用媒体查询，在浏览器窗口逐渐缩小时，设置表单的对应样式，重点设置<input>控件的宽度。

（4）在 div.element-right 中嵌套 div.teddy-text 和 div.teddy-follow。其中，在 div.teddy-text 中添加头像和昵称，在 div.teddy-follow 中添加粉丝量和关注量，对元素设置宽度时，尽量使用百分比，这样可以使元素自动响应视口大小的变化。

（5）在 div.element-last 中使用 ul 列表添加菜单部分，在每个<li>标签中嵌套<a>标签和标签，其中标签用于设置菜单的图标。

【代码实现】

对页面的结构有所了解后，接下来我们编写代码来实现它。该项目的 HTML 代码如 code\07\0702\0702.html 所示。

0702.html

```
<!DOCTYPE HTML>
<html lang="en">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1,
        maximum-scale=1">
    <title>个人信息页面</title>
    <link href="css/personal_info.css" rel="stylesheet" type="text/css"
        media="all"/>
</head>
<body>
    <h1>个人信息</h1>
    <div class="element">
        <div class="element-left">
            <div class="edit">
```

```
                <form>
                    <ul>
                        <li>
                            <input type="text" placeholder="请输入旧密码">
                            <a href="#" class="user1"></a>
                        </li>
                        <li>
                            <input type="text" placeholder="请输入新密码">
                            <a href="#" class="lock1"></a>
                        </li>
                    </ul>
                </form>
                <div class="submit">
                    <input type="submit"  value="修改密码" >
                </div>
            </div>
        </div>
        <div class="element-right">
            <div class="teddy-bear">
                <div class="teddy-text">
                    <img src="images/b1.jpg" class="img-responsive">
                    <h4>科学怪人</h4>
                </div>
                <div class="teddy-follow">
                    <ul>
                        <li class="folw-h">
                            <h3>2.5 万</h3>
                            <p>粉丝</p>
                        </li>
                        <li class="folw-r">
                            <h3>100</h3>
                            <p>关注</p>
                        </li>
                        <div class="clear"></div>
                    </ul>
                </div>
            </div>
        </div>
        <div class="element-last">
            <ul class="content">
              <li class="menu button">
                <a href="#">个人资料<span class="icon"></span></a>
              </li>
              <li class="menu button">
            <a href="#">消息<span class="icon stat"></span><i>13</i></a>
              </li>
              <li class="menu button">
                  <a href="#">设置<span class="icon msg"></span></a>
              </li>
              <li class="menu info button">
```

```
                <a href="#">退出登录<span class="icon signout"></span></a>
            </li>
        </ul>
    </div>
    <div class="clear"> </div>
  </div>
</body>
</html>
```

该项目的 CSS 代码如 code\07\0702\css\ personal_info.css 所示。

personal_info.css

```
/*第 7 单元 项目 7-2 社交网站个人信息页面 CSS 文件*/
*{margin:0;padding:0;border:0;}
ul{list-style:none;}
body {
    font-family: 'Roboto', sans-serif;
    font-size: 100%;
    font:inherit;/*继承父元素的字体系列样式*/
    background: url(../images/2.png) repeat 0px 0px;
}
a {
    text-decoration: none;
}
a:hover {
    transition: 0.5s all;
    -webkit-transition: 0.5s all;

}
.clear{
    clear:both;/*清除左右浮动*/
}
/*--header--*/
h1 {
    font-family: '微软雅黑';
    font-size: 3em;
    color: #3C2F24;
    text-align: center;
    margin: 2em 0em 1em 0em;
}
/*--element--*/
.element {
    width: 60%;
    margin: 0 auto 7em;
    padding: 2em;
    display: flex;/*设置该 div 为一个弹性盒容器*/
    flex-flow: row;/*子元素按横轴方向顺序排列*/
}
.element-left {
    margin:0 1.5%;
```

```
    flex: 3.3 ;/*将容器分为 10 份，占 1/3*/
    order: 1;/*排序为第 1 个子元素*/
}
.element-right {
    margin:0 1.5%;
    flex: 3.3 ;/*将容器分为 10 份，占 1/3*/
    order:2;/*排序为第 2 个子元素*/
}
.element-last {
    margin:0 1.5%;
    flex: 3.3 ;/*将容器分为 10 份，占 1/3*/
    order:3;/*排序为第 3 个子元素*/
}
/*--element-left--*/
.edit {
    margin:0 auto;/*外边距：上下 0，左右水平居中*/
    background: #fff;
}
form {
    padding: 10% 4% 8% 4%;
}
form li {
    list-style: none;
    width: 100%;
    font-weight: 500;
    border: 1px solid #ccc;
    background: #fff;
    margin: 1em 0;
}
input[type="text"] {
    width: 83%;
    padding:1em 0em 1em 1.7em;
    font-size: 15px;
    font-weight: 500;
    color: #858282;
}
.user1 {
    width: 20px;
    height: 20px;
    display: block;
    float: left;
    margin: 12px 0px 0px 0px;
    border-right: 1px solid #999;
    padding: 5px 16px 0 0;
    background: url(../images/user.png) no-repeat 8px 2px ;
}
.lock1 {
    width: 20px;
    height: 20px;
    display: block;
```

```
    float: left;
    margin: 12px 0px 0px 0px;
    border-right: 1px solid #999;
    padding: 5px 16px 0 0;
    background: url(../images/lock.png) no-repeat 8px 2px ;
}
.submit input[type="submit"]{
    width: 100%;
    padding:21px 20px;
    background: #ef8d32;
    font-size:20px;
    font-weight: 400;
    color: #fff;
    cursor: pointer;
    transition: 0.1s all;
    -webkit-transition: 0.1s all;
}
input[type="submit"]:hover{
    background:#ee5a32;
}
/*--element-right--*/
.teddy-bear {
    text-align: center;
}
.teddy-text {
    background: #fff;
    padding: 1em 1em;
}
.teddy-text h4 {
    font-size: 1.2em;
    color: #ee5a32;
    margin: 0.5em 0 0;
}
.teddy-text img {
    width: 60%;
    padding: 8px;
    border: 3px solid #ee5a32;
    border-radius: 70px;/*边框圆角半径为 70px*/
}
.teddy-follow {
    background: #ef8d32;
    padding: 0.7em 0em 0.7em 0em;
}
.teddy-follow li {
    display: inline-block;
    width: 49%;
    float: left;
    text-align: center;
}
.teddy-follow li h3 {
```

```
    font-size: 0.95em;
    font-weight: bold;
    color: #fff;
}
.teddy-follow li p {
    font-size: 0.8em;
    color: #fff;
}
.teddy-follow li.folw-h {
    border-right: 1px solid #fff;
}
/*--element-last--*/
ul.content {
    width: 100%;
    margin: 0 auto;
    background: #ef8d32;
    font-weight: 100;
}
li.button a {
    padding: 23.5px 27px;
    margin: 0;
    display: block;
}
li.button {
    list-style: none;
    text-align: left;/*文本对齐方式为居左对齐*/
}
li.menu{
    width: 100%;
    margin: 0;
    padding: 0;
    border-bottom: 1px solid #CD5F4A;
}
li.menu:hover {
    background: #ee5a32;
}
li.button a:hover{
    text-decoration:none;
}
li.button a span{
    margin-right: 22px;
}
li.menu.info {
    border-bottom: none;
}
li.button i {
    display: block;
    float: right;
    padding: 4px 12px;
    border-radius: 4px;
```

```
    -webkit-border-radius: 4px;
    background: #505771;
    font-size: 13px;
    font-style: normal;/*设置字体样式正常，因为<i>标签表示文本字体为斜体*/
}
a, a:visited {
    color:#fff;
}
.icon{
    width: 25px;
    height: 23px;
    display: inline-block;
    float:left;
}
.icon {
    background: url(../images/img-sprite.png) no-repeat -1px -2px;
}
.stat{
    background: url(../images/img-sprite.png) no-repeat -30px -4px;
}
.msg{
    background: url(../images/img-sprite.png) no-repeat -62px -2px;
}
.signout{
    background: url(../images/img-sprite.png) no-repeat -93px -2px;
}
/*--媒体查询--*/
@media(max-width:1366px){
    input[type="text"] {
        padding: 1em 0em 1em 0.7em;
    }
    .user1 {
        margin: 12px -3px 0px 0px;
        padding: 5px 10px 0 0;
    }
    .lock1 {
        margin: 12px -3px 0px 0px;
        padding: 5px 10px 0 0;
    }
}
@media(max-width:1280px){
    input[type="text"] {
        width: 75%;
    }
    .element {
        width: 82%;
        margin: 0 auto 6em;
    }
    form {
        padding: 12% 4% 12% 4%;
```

```
239    }
240}
241@media(max-width:768px){
242    li.button a {
243        padding: 23.5px 16px;
244    }
245    input[type="text"]{
246        width: 60%;
247    }
248    .teddy-text h4 {
249        font-size: 1em;
250    }
251    li.button a {
252        padding: 22.5px 10px;
253    }
254}
255@media (max-width: 667px){
256    input[type="text"] {
257        width: 70%;
258    }
259    form {
260        padding: 13% 6% 13% 6%;
261    }
262    .teddy-text h4 {
263        font-size: 0.92em;
264    }
265}
266@media (max-width: 640px){
267    .element{
268        margin: 0 auto 2em;
269        flex-flow: column; /*弹性盒中的子元素按纵轴方向排列*/
270    }
271    .element-left,.element-right,.element-last{
272        order: 0;/*去除顺序依次纵向排列*/
273        margin: 1%;
274    }
275    h1 {
276        font-size: 2em;
277        margin: 1em 0em 0.5em 0em;
278    }
279    form {
280        padding: 13% 13% 13% 13%;
281    }
282    input[type="text"] {
283        width: 89%;
284    }
285}
286@media (max-width: 600px){
287    input[type="text"] {
288        width: 80%;
```

```
289    }
290}
291@media (max-width: 568px){
292    input[type="text"] {
293        width: 70%;
294    }
295}
```

【项目总结】

本项目的练习重点：

通过本项目的练习，读者应该学会使用弹性盒布局来制作响应式网站。

本项目的练习方法：

（1）用弹性盒布局做出整体结构，参考 demo7-5.html。

（2）向弹性盒的子元素添加结构内容。

（3）给结构添加样式。

（4）用媒体查询进行调整。

本项目的注意事项：

弹性盒布局在移动端的出现率非常高，建议读者深刻理解【前导知识】中的弹性盒布局案例，再开始本项目的编码。

【思考题】

1. 请简述什么是视口，以及 PC 端是否存在视口。
2. 请简述响应式 Web 设计中为什么要使用百分比布局。

Responsive Web Design

8 Chapter

单元 8
响应式设计神器——Bootstrap

在前面的单元中讲解了响应式 Web 设计，相信会有一部分读者觉得实现响应式是个很复杂的过程，需要调整很多细节。这时，编程人员需要一个工具——Bootstrap。它为大多数标准的 UI 设计场景提供了用户友好、跨浏览器的解决方案，让响应式变得容易实现，大大提高了前端开发的工作效率。本单元将针对如何使用 Bootstrap 进行响应式 Web 设计进行详细的讲解。

【教学导航】

学习目标	能够使用 Bootstrap 制作响应式网站，具体包括： 1. 掌握 Bootstrap 的安装； 2. 掌握 Bootstrap 的布局容器； 3. 掌握使用 Bootstrap 制作表单和按钮； 4. 掌握使用 Bootstrap 制作导航栏、轮播事件、标签页； 5. 熟悉 Bootstrap 中的栅格系统
教学方式	本单元将通过一个项目让读者掌握 Bootstrap 的使用，在完成项目 8-1 的过程中，需要注意的是： 时刻思考响应式 web 设计的理念； 做到对 Bootstrap 提供的类了然于心，用时方便查找； 学会覆盖 Bootstrap 提供的样式
重点知识	1. Bootstrap 组件的使用 2. 如何将各模块组合成完整的网页
关键词	Bootstrap、container、navbar、form、btn、col-*-*、nav-tabs、响应式工具、carousel

【项目 8】Bootstrap 餐饮类网站首页

【项目描述】

目前电商网站层出不穷，由于移动设备方便携带，移动端用户人数倍增，逐渐超过 PC 端，所以众电商们十分注重网站的响应式设计，特别是在移动端也能提供良好的用户体验。本单元将带领读者实现一个响应式的餐饮类电商网站，相信这些美食可以吸引你的眼球！该网页上半部分效果如图 8-1 所示。

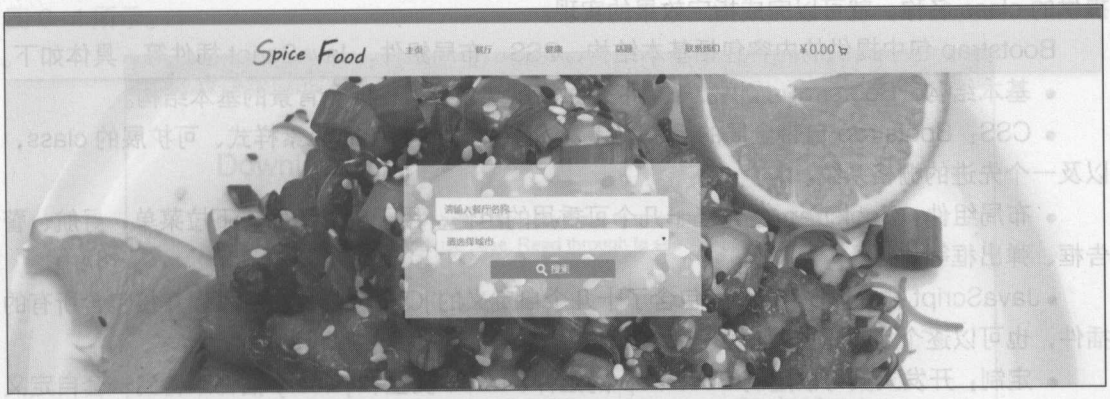


图8-1　首页展示效果—上半部分

用鼠标向下滑动页面可以看到网页下半部分，如图 8-2 所示。

网页是由多个模块组成的，本项目模块分为 header 部分、搜索区域、热卖商品、轮播广告、特色推荐、footer 部分，为了让读者更清楚本项目的实现效果，后面将按网页中的模块分成几个任务，带领读者一步一步地完成该项目。

图8-2　首页展示效果—下半部分

【任务 1-完成网页 header 部分】

【前导知识】

Bootstrap 简介

Bootstrap 是由 Twitter（著名的社交网站）推出的前端开源工具包，它基于 HTML、CSS、JavaScript 等前端技术，2011 年 8 月在 GitHub 上发布，一经推出就颇受欢迎。在本书编著时 Bootstrap 的最新版本是 3.3.6。

Bootstrap 中预定义了一套 CSS 样式和与样式对应的 jQuery（jQuery 是一个快速、小巧、功能丰富的 JavaScript 库）代码，应用时我们只需提供固定的 HTML 结构，添加 Bootstrap 中提供的 class 名称，就可以完成指定效果的实现。

Bootstrap 包中提供的内容包括基本结构、CSS、布局组件、JavaScript 插件等，具体如下。

- 基本结构：Bootstrap 提供了一个带有网格系统、链接样式、背景的基本结构。
- CSS：Bootstrap 自带全局的 CSS 设置、定义基本的 HTML 元素样式、可扩展的 class，以及一个先进的栅格系统。
- 布局组件：Bootstrap 包含了十几个可重用的组件，用于创建图像、下拉菜单、导航、警告框、弹出框等。
- JavaScript 插件：Bootstrap 包含了十几个自定义的 jQuery 插件。它可以直接包含所有的插件，也可以逐个包含这些插件。
- 定制：开发人员可以定制 Bootstrap 的组件、LESS 变量和 jQuery 插件来得到一套自定义的版本。

Bootstrap 之所以受到广大前端开发人员的欢迎，是因为使用 Bootstrap 可以构建出非常优雅的前端界面，而且占用资源非常小，另外 Bootstrap 还具有以下几个优势：

- 移动设备优先：自 Bootstrap 3 起，移动设备优先的样式贯穿于整个库。
- 浏览器支持：主流浏览器都支持 Bootstrap，包括 IE、Firefox、chrome、Safari 等。

- 容易上手：要学习 Bootstrap，只需读者具备 HTML 和 CSS 的基础知识。
- 响应式设计：Bootstrap 的响应式 CSS 能够自适应于台式机、平板电脑和手机的屏幕大小。
- 良好的代码规范：为开发人员创建接口提供了一个简洁、统一的解决方案，减少了测试的工作量；使开发人员站在巨人的肩膀上，不重复造轮子。
- 组件：Bootstrap 包含了功能强大的内置组件。
- 定制：Bootstrap 还提供了基于 Web 的定制。

Bootstrap 下载和环境安装

1. 下载

学习任何工具框架都会涉及下载的问题，Bootstrap 也不例外。首先，打开浏览器，访问 Bootstrap 官方网站地址“http://getbootstrap.com/”，下载 Bootstrap 的最新版本，如图 8-3 所示。读者也可以直接使用本教材的源代码中下载好的 bootstrap-3.3.6-dist.zip。

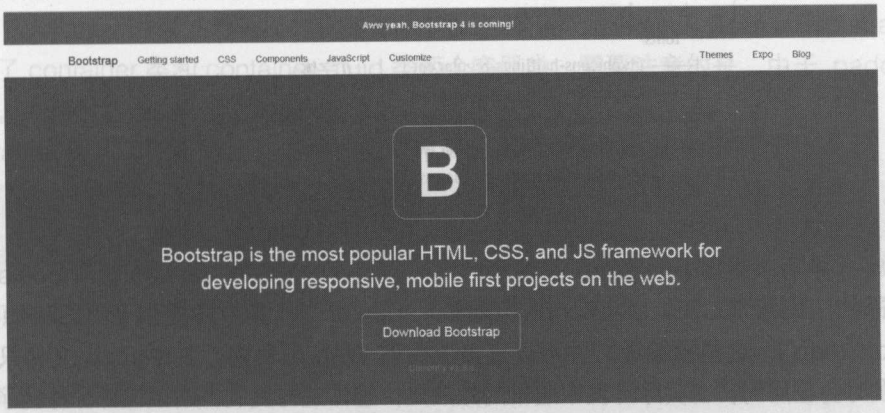


图8-3　Bootstrap官网首页

单击“Download Bootstrap”按钮，跳转至下载页面，将页面下拉会看到 3 个按钮，如图 8-4 所示。

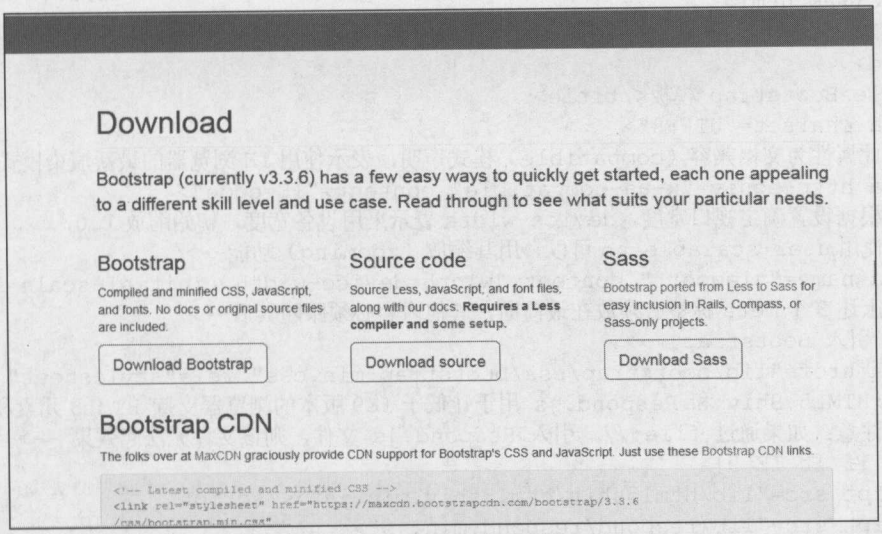


图8-4　Bootstrap下载页面

其中，“Download Bootstrap”按钮用于下载 Bootstrap。单击该按钮，可以下载 Bootstrap CSS、JavaScript 和字体的预编译的压缩版本，不包含文档和最初的源代码文件。下载成功后，解压缩 ZIP 文件，将看到下面的文件和目录结构，如图 8-5 所示。

```
bootstrap/
├── css/
|   ├── bootstrap.css                    // 预定义的 CSS 文件
|   ├── bootstrap.css.amp                //CSS 与 less 源码对应文件
|   ├── bootstrap.min.css                // 压缩
|   ├── bootstrap.min.css.map
|   ├── bootstrap-theme.css              // 主题文件
|   ├── bootstrap-theme.css.map
|   ├── bootstrap-theme.min.css
|   └── bootstrap-theme.min.css.map
├── js/
|   ├── bootstrap.js                     //js
|   └── bootstrap.min.js
└── fonts/
    ├── glyphicons-halflings-regular.eot // 字体
    ├── glyphicons-halflings-regular.svg
    ├── glyphicons-halflings-regular.ttf
    ├── glyphicons-halflings-regular.woff
    └── glyphicons-halflings-regular.woff2
```

图8-5　Bootstrap目录结构

在图 8-5 中可以看出 bootstrap 的基本结构，预编译的 CSS、JavaScript（bootstrap.*），以及预编译的压缩版 CSS、JavaScript（bootstrap.min.*），其中还提供了 CSS 源码映射表（bootstrap.*.map），这些预编译文件可以直接应用到 Web 项目中，其中 map 文件只有在自定义高级开发时会应用到，在实际开发中通常的做法是，整体进行复制，所以该部分作为了解即可。同时，Bootstrap 包中还包含了 Glyphicons 的字体文件，在附带的 Bootstrap 主题中使用了这些图标。

2. 环境安装

一个使用了 Bootstrap 的基本的 HTML 模板如下所示。

```
<!DOCTYPE html>
<html>
<head>
<title>Bootstrap 模板</title>
<meta charset="UTF-8">
<!--此属性为文档兼容（compatible）模式声明，表示使用 IE 浏览器的最新渲染模式-->
<meta http-equiv="x-ua-compatible" content="IE=edge">
<!--根据设置确定视口宽度，device-width 表示采用设备宽度，初始缩放 1.0，
    使用 user-scalable=no 可以禁用其缩放（zooming）功能-->
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<!--上述 3 个 meta 标签必须放在最前面，其他内容必须跟随其后-->
<!-- 引入 Bootstrap -->
<link href="lib/bootstrap/css/bootstrap.min.css" rel="stylesheet">
<!-- HTML5 Shiv 和 Respond.js 用于让低于 IE9 版本的浏览器支持 HTML5 元素和媒体查询
    注意：如果通过 file:// 引入 Respond.js 文件，则该文件无法起效果 -->
<!--[if lt IE 9]>
<script src="lib/html5shiv/html5shiv.min.js"></script>
<script src="lib/respond/respond.min.js">
</script>
```

```
<![endif]-->
</head>
<body>
<!-- jQuery (Bootstrap 的 JavaScript 插件需要引入 jQuery) -->
<script src="lib/jquery/jquery-1.11.0.min.js"></script>
<!-- 包括所有已编译的插件 -->
<script src="lib/bootstrap/js/bootstrap.min.js"></script>
</body>
</html>
```

如果想在一个 HTML 文件中使用 Bootstrap，该文件必须引入包 jquery.js、bootstrap.min.js 和 bootstrap.min.css 文件，在上述代码中 3 个<meta>标签分别用于设置字符集、文档兼容模式和视口，html5shiv.min.js 用于使 IE8 支持 HTML5 元素，Respond.js 用于使 IE8 支持媒体查询。

布局容器

使用 Bootstrap 时需要为页面内容和栅格系统包裹一个.container 容器。Bootstrap 包中为我们提供了.container 类和.container-fluid 类两个容器类。需要注意的是，由于 padding 等属性的原因，这两种容器类不能互相嵌套。

.container 类用于固定宽度并支持响应式布局的容器，示例代码如下所示。

```
<div class="container">
  ...
</div>
```

.container-fluid 类用于设置 100%宽度，占据全部视口（viewport）的容器，示例代码如下所示。

```
<div class="container-fluid">
  ...
</div>
```

两种容器在页面中使用的对比效果如 demo8-1 所示。

demo8-1.html

```
<!DOCTYPE html>
<html>
<head>
    <title>布局容器</title>
    <meta charset="UTF-8">
    <meta http-equiv="x-ua-compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="lib/bootstrap/css/bootstrap.min.css" rel="stylesheet">
</head>
<br>
<!--居中显示，两边有留白-->
<div class="container" style="border:1px solid #000000;
      height:100px;">.container</div>
<!--整个宽度-->
<div class="container-fluid" style="border:1px solid #000000;
      height:100px;">.container-fluid</div>
<script src="lib/jquery/jquery-1.11.0.min.js"></script>
```

```
16 <script src="lib/bootstrap/js/bootstrap.min.js"></script>
17 </body>
18 </html>
```

用浏览器打开 demo8-1，页面效果如图 8-6 所示。

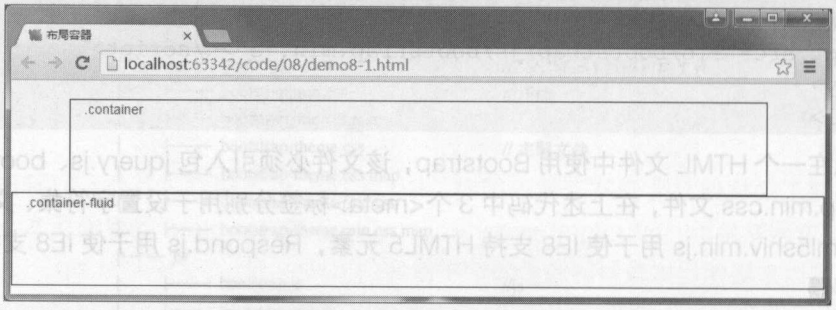


图8-6　demo8-1页面效果

从图 8-6 中的页面效果可以看出，使用.container 容器布局时页面两边有留白，而使用. container-fluid 容器布局时会占用页面的整个宽度。

导航栏（navbar）

Bootstrap 导航栏是在应用或网站中作为导航页头的响应式基础组件。Bootstrap 中为我们提供了默认样式的导航条，它在移动设备上可以折叠（并且可开可关），且在视口（viewport）宽度增加时逐渐变为水平展开模式，如图 8-7 所示。

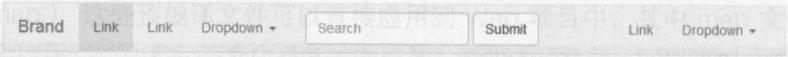


图8-7　Bootstrap提供的默认导航条

缩小浏览器窗口后，菜单均被隐藏，代替出现的是一个汉堡按钮，如图 8-8 所示。

单击图 8-8 所示的汉堡按钮“≡”，显示被隐藏菜单的下拉列表，如图 8-9 所示。

图8-8　汉堡按钮

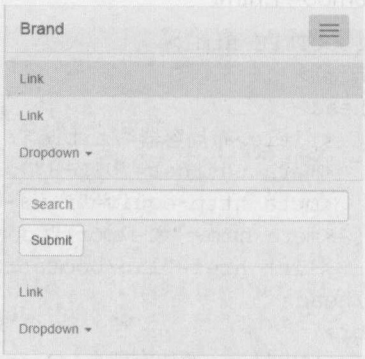


图8-9　被隐藏的菜单列表

由于 Bootstrap 导航栏在实际应用中比较复杂，本书中主要为读者介绍如何使用 Bootstrap 制作一个基础导航栏，以及如何修改默认导航栏的样式。

1. 基础导航栏

使用 Bootstrap 制作基础导航栏主要分为以下步骤。

（1）添加一个容器<nav>或 标签，使用.navbar 类和.navbar-default 类，并且添加role="navigation"，增加可访问性。

（2）向 标签添加一个标题，使用.navbar-header 类，内部包含带有.navbar-brand 类的<a>标签，用于定义品牌图标，如果是文字视觉上会稍大些。

（3）要向导航栏添加链接，只需要简单地添加带有.nav 类、.navbar-nav 类的无序列表即可。

基础导航栏的具体实现代码如 demo8-2 所示。

demo8-2.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Bootstrap 基础导航栏</title>
    <meta charset="UTF-8">
    <meta http-equiv="x-ua-compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="lib/bootstrap/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
<nav class="navbar navbar-default" role="navigation">
    <!--这里可以定义品牌图标-->
    <div class="navbar-header">
        <a class="navbar-brand " id="nav-brand-itheima" href="#" >
            网站首页
        </a>
    </div>
    <ul class="nav navbar-nav">
        <li class="active"><a href="##">系列教程</a></li>
        <li><a href="##">名师介绍</a></li>
        <li><a href="##">成功案例</a></li>
        <li><a href="##">关于我们</a></li>
    </ul>
</nav>
<script src="lib/jquery/jquery-1.11.0.min.js"></script>
<script src="lib/bootstrap/js/bootstrap.min.js"></script>
</body>
</html>
```

用浏览器打开 demo8-2，页面效果如图 8-10 所示。

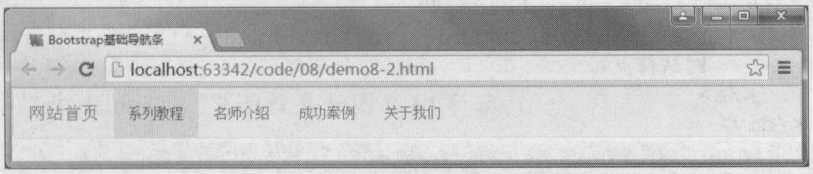


图8-10　demo8-2页面效果

2. 实现响应式

基础导航栏只能适应大屏幕的浏览器，当浏览器窗口缩小到一定程度时，菜单将被折叠，如图 8-11 所示。

实现菜单折叠的效果有以下两个步骤。

（1）实现菜单的折叠和隐藏，把小屏幕显示时需要折叠的内容包裹在一个 标签内，并且为这个 标签使用.collapse、.navbar- collapse 两个类，最后为这个 div 添加一个 id。

（2）添加在小屏幕时，要显示的汉堡按钮的固定写法如下。

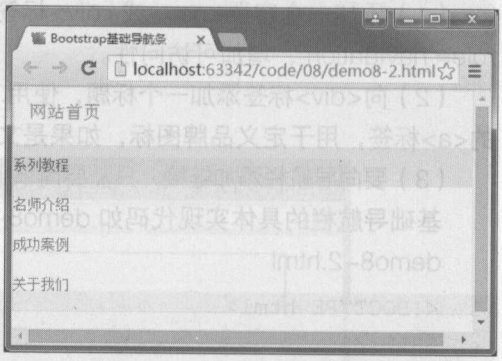


图8-11　demo8-2菜单折叠效果

```
    <button class="navbar-toggle" type= "button"
data-toggle="collapse">
     <span class="sr-only">Toggle Navigation
</span>
     <span class="icon-bar"></span>
     <span class="icon-bar"></span>
     <span class="icon-bar"></span>
    </button>
```

完整的响应式基础导航栏的实现代码如 demo8-3 所示。

demo8-3.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Bootstrap 响应式导航栏</title>
    <meta charset="UTF-8">
    <meta http-equiv="x-ua-compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="lib/bootstrap/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
<nav class="navbar navbar-default" role="navigation">
        <button type="button" class="navbar-toggle collapsed"
                data-toggle= "collapse" data-target="#navbar-collapse"
                aria-expanded="false">
            <span class="sr-only">汉堡按钮</span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
        </button>
    <!--这里可以定义品牌图标-->
        <div class="navbar-header">
            <a class="navbar-brand " id="nav-brand-itheima" href="#" >
                网站首页
            </a>
        </div>
    <div class="collapse navbar-collapse" id="navbar-collapse">
        <ul class="nav navbar-nav">
            <li class="active"><a href="##">系列教程</a></li>
            <li><a href="##">名师介绍</a></li>
            <li><a href="##">成功案例</a></li>
            <li><a href="##">关于我们</a></li>
        </ul>
```

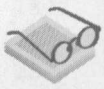
```
</div>
</nav>
<script src="lib/jquery/jquery-1.11.0.min.js"></script>
<script src="lib/bootstrap/js/bootstrap.min.js"></script>
</body>
</html>
```

用浏览器打开 demo8-3，浏览器窗口缩小的效果如图 8-12 所示。

在 demo8-3 中，为折叠菜单添加的 id 值为 navbar-collapse。在<button>标签添加 data-target="#navbar-collapse"，代表这个按钮控制的是 id 值为 navbar-collapse 的容器。单击汉堡按钮“☰”，即可显示下拉菜单，如图 8-13 所示。

图8-12　demo8-3小屏幕页面效果

图8-13　下拉菜单



多学一招：如何修改 Bootstrap 默认样式

Bootstrap 为我们提供的是基础的 CSS 样式，如果想自定义样式，有两种方式。

（1）最直接的方式是查找 Bootstrap 源码样式，用 CSS 覆盖掉这些默认样式。具体可以通过查看针对 Bootstrap 中使用的 demo 使用哪些类名控制我们要修改的样式，然后使用该类名并且编写自己的样式来实现覆盖。

例如，修改导航条的默认背景色可以通过为.navbar-default 类添加样式来实现，在 demo8-3 中添加如下代码。

```
<style>
  .navbar-default{
        background-color: pink;
    }
</style>
```

用浏览器打开 demo8-3，页面效果如图 8-14 所示。

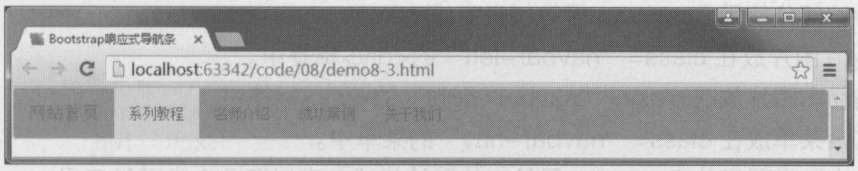


图8-14　导航条背景色的变化效果

（2）使用!important 提高代码优先级。读者可自行尝试。

【任务描述】

学习了 Bootstrap 的布局容器和导航栏，就可以完成本单元项目的 header 部分了。该项目 header 部分的页面效果如图 8-15 所示。

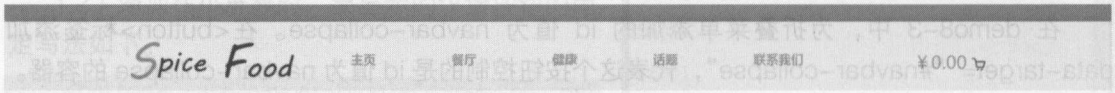


图8-15　header部分的页面效果

浏览器窗口缩小至小屏幕（<992px）时，logo 换行的页面效果如图 8-16 所示。

浏览器窗口缩小至超小屏幕（<768px）时，出现汉堡按钮，如图 8-17 所示。

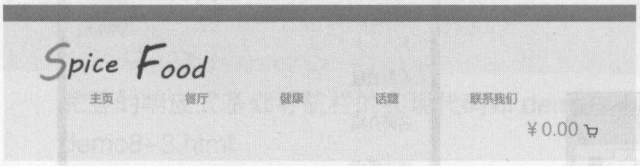


图8-16　小屏幕页面效果

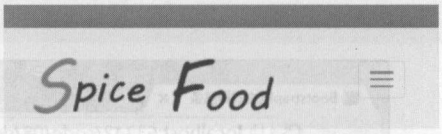


图8-17　超小屏幕页面效果

单击汉堡按钮显示下拉菜单，如图 8-18 所示。

【任务分析】

了解了该任务要实现的效果后，接下来我们分析一下页面结构，如图 8-19 所示。

图8-18　超小屏幕下拉菜单

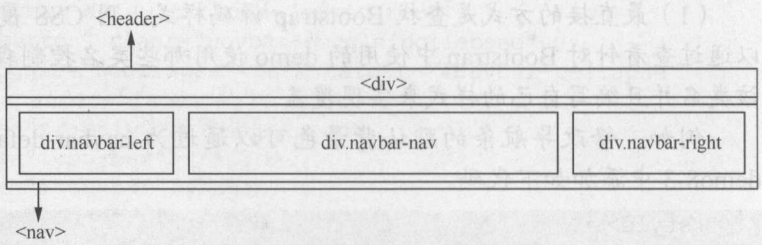


图8-19　header头部结构图

图 8-19 所示的导航页面主要由两部分构成，即添加背景色的 标签（绿色横条）和<nav>标签。为了更好地体现 HTML5 的语义化，整个 header 部分所有内容应该包含在一个<header>标签中。该部分的实现细节具体分析如下。

（1）导航部分使用 Bootstrap 导航栏来完成。

（2）logo 图片放在 class=“navbar-left”的 标签中。

（3）购物车图片放在 class=“navbar-right”的 标签中。

（4）其余菜单放在 class=“navbar-nav”的菜单中。

另外，这里需要覆盖 Bootstrap 导航栏的默认样式，有以下几个类可供使用。

（1）.navbar-brand：通过该类设置 logo 部分图片的宽高、位置等。

（2）.navbar-default、.navbar-nav：通过这两个类设置导航菜单的样式，如鼠标悬停变色的效果等。

（3）.navbar-right：通过该类设置导航右侧购物车部分的样式。

最后将所有的字体统一设置成微软雅黑。

【代码实现】

对页面的结构有所了解后，即可开始编写代码来实现它。该任务的代码如 code\08\0801\0801_1.html 所示。

0801_1.html

```
<!DOCTYPE html>
<html>
<head>
    <title>餐饮网站 header 部分</title>
    <meta charset="UTF-8">
    <meta http-equiv="x-ua-compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- 引入 Bootstrap -->
    <link href="lib/bootstrap/css/bootstrap.min.css" rel="stylesheet">
    <style type="text/css">
        /*设置 body 元素字体为 Microsoft YaHei*/
        body { font-family: Microsoft YaHei}
        .head-top{
            background: #5fa022;
            padding: 0.8em 0;
        }
        .navbar-brand{
            padding: 0 0;
        }
        /*设置图片 logo 的大小和位置*/
        .navbar-brand>img{
            height: auto;
            margin-right: 5px;
            margin-top: 5px;
            width: 250px;
        }
        /*设置整个导航菜单的内边距、背景色和阴影*/
        .navbar-default{
            padding: 1.5em 0;
            background-color: #f2f0f1;
            box-shadow: 12px -5px 39px -12px;
        }
        /*设置导航栏中菜单 a 链接的样式*/
        .navbar-default .navbar-nav>li a {
            top: 10px;
            padding: 0.5em 3em;
            text-decoration: none;
```

```
            font-weight: 600;
            font-size: 1.2em;
            color:#919191;
        }
        /*设置导航栏菜单的鼠标悬停和获取焦点时的状态*/
        .navbar-default .navbar-nav> li >a:hover,
        .navbar-default .navbar-nav> li >a:focus {
            background:#D96B66;
            color: white;
            border-radius: 3px;
            -webkit-border-radius: 3px;
        }
        /*设置导航栏右侧 a 链接（0.00）的颜色*/
        .navbar-right>a{
            color:#D96B66;
        }
        /*媒体查询：当视口小于 970px 时，导航菜单字体变小，避免了换行的问题*/
        @media (max-width: 970px) {
            .navbar-default .navbar-nav>li a {
                font-size: 1.1em;
            }
        }
        /*header 部分结束*/
    </style>
</head>
<body>
<!-- header -->
<header>
    <div class="head-top"></div>
    <nav class="navbar navbar-default" >
        <div class="container-fluid">
            <div class="container">
                <div class="navbar-header">
                    <!--<div class="logo">-->
   <a href="#" class="navbar-brand" ><img src="images/logo.png"  alt=""></a>
                    <!--</div>-->
    <!-- 当进入最小设备时，导航条将隐藏，显示汉堡按钮，单击时可以切面显示导航条隐藏的信
          息   * data-target : 用于确定需要显示 div   -->
          <button type="button" class="navbar-toggle collapsed"
                  data-toggle= "collapse" data-target="#navbar-collapse"
                  aria-expanded="false">
                    <span class="sr-only">汉堡按钮</span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
            </div>
            <!-- 导航链接、表单和其他内容切换 -->
            <div class="collapse navbar-collapse" id="navbar-collapse">
                <ul class="nav navbar-nav">
```

```
                    <li><a href="#">主页</a></li>
                    <li><a href="#">餐厅</a></li>
                    <li><a href="#">健康</a></li>
                    <li><a href="#">话题</a></li>
                    <li><a href="#">联系我们</a></li>
                </ul>
                <div class="navbar-right ">
                    <a href="#">
        <h3> <span >¥0.00 <img src="images/bag.png" alt=""/></span></h3>
                    </a>
                </div>
            </div>
          </div>
        </div><!-- /.navbar-collapse -->
      </div><!-- /.container-fluid -->
  </nav>
</header>
<!-- header -->
<!-- jQuery (Bootstrap 的 JavaScript 插件需要引入 jQuery) -->
<script src="lib/jquery/jquery-1.11.0.min.js"></script>
<!-- 包括所有已编译的插件 -->
<script src="lib/bootstrap/js/bootstrap.min.js"></script>
</body>
</html>
```

【任务 2-完成网页搜索模块】

【前导知识】

表单

几乎所有的网站中都涉及表单的应用，接下来我们将介绍如何使用 Bootstrap 创建表单。

Bootstrap 通过一些简单的 HTML 标签和扩展的类即可创建出不同样式的表单，按照布局的不同，表单主要分为垂直表单（默认）、内联表单和水平表单 3 类。

1. 垂直表单

垂直表单也称为基本表单，使用 Bootstrap 制作基本表单主要分为以下步骤。

（1）向父<form>标签添加 role=“form”。

（2）把标签和控件放在一个类名为 form-group 的 中，获取最佳间距。

（3）向所有的文本标签<input>、<textarea>和<select>添加.form-control 类。

垂直表单的具体实现如 demo8-4 所示。

demo8-4.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Bootstrap 基本表单</title>
    <meta charset="UTF-8">
    <meta http-equiv="x-ua-compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
    <link href="lib/bootstrap/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
<form role="form">
    <div class="form-group">
        <label for="name">名称</label>
        <input type="text" class="form-control" id="name"
               placeholder="请输入名称">
    </div>
    <div class="form-group">
        <label for="inputfile">文件输入</label>
        <input type="file" id="inputfile">
        <p class="help-block">这里是块级帮助文本的实例</p>
    </div>
    <div class="checkbox">
        <label>
            <input type="checkbox">记住我
        </label>
    </div>
    <button type="submit" class="btn btn-default">提交</button>
</form>
</body>
</html>
```

用浏览器打开 demo8-4，页面效果如图 8-20 所示。

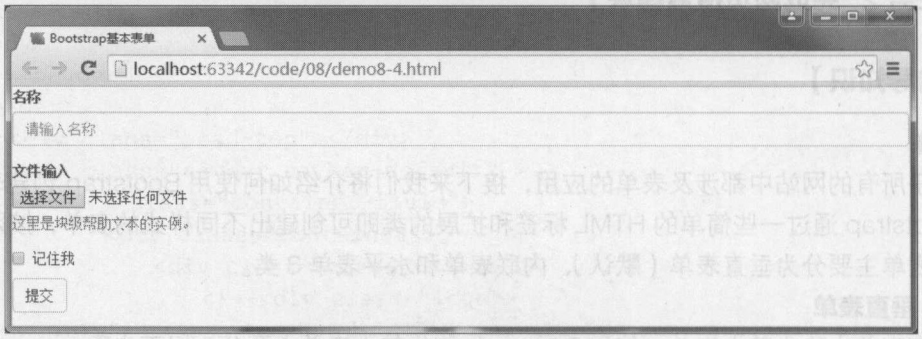


图8-20　demo8-4页面效果

2. 内联表单

Bootstrap 创建内联表单，只需要在垂直表单的基础上，为<form>标签添加类.form-inline，具体代码如 demo8-5 所示。

demo8-5.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Bootstrap 内联表单</title>
    <meta charset="UTF-8">
    <meta http-equiv="x-ua-compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="lib/bootstrap/css/bootstrap.min.css" rel="stylesheet">
```

```
</head>
<body>
<form role="form" class="form-inline">
   <div class="form-group">
      <label for="name">名称</label>
      <input type="text" class="form-control" id="name"
           placeholder="请输入名称" width="50px">
   </div>
   <div class="form-group">
      <label for="inputfile">文件输入</label>
      <input type="file" id="inputfile">
      <p class="help-block">这里是块级帮助文本的实例</p>
   </div>
   <div class="checkbox">
      <label>
         <input type="checkbox" class="sr-only">记住我
      </label>
   </div>
   <button type="submit" class="btn btn-default">提交</button>
</form>
</body>
</html>
```

用浏览器打开 demo8-5，页面效果如图 8-21 所示。

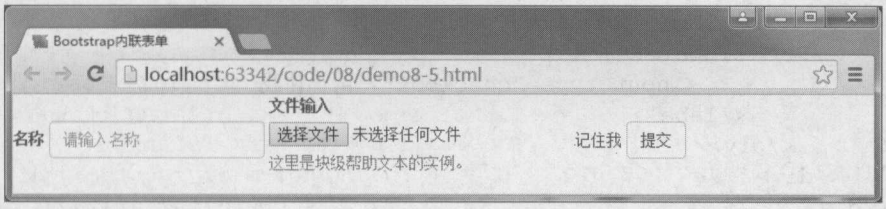


图8-21　demo8-5页面效果

从图 8-15 中的页面效果可以看出，内联变淡的所有元素是内联的，向左对齐的，标签是并排的，默认情况下，Bootstrap 中的<input>、<select>和<textarea>标签有 100%宽度。在使用内联表单时，读者需要自己在表单控件上设置一个宽度，可以使用类.sr-only 隐藏内联表单的某个标签。

3. 水平表单

水平表单与其他表单不仅在标记的数量上不同，而且表单的呈现形式也不同。创建水平布局的表单的步骤具体如下。

（1）向父<form>标签添加类.form-horizontal，改变.form-group 的行为，并使用 Bootstrap 预置的栅格 class 将 label 和控件组水平并排布局。

（2）把标签和控件放在一个带有.form-group 类的 中。

（3）向标签添加.control-label 类。

实现水平表单的具体代码如 demo8-6 所示。

demo8-6.html

```
<!DOCTYPE html>
<html>
<head>
```

```
    <title>Bootstrap 水平表单</title>
    <meta charset="UTF-8">
    <meta http-equiv="x-ua-compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="lib/bootstrap/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
<form class="form-horizontal" role="form">
    <div class="form-group">
        <label for="name" class="col-sm-2 control-label">用户名</label>
        <div class="col-sm-10">
            <input type="text" class="form-control" id="name"
                   placeholder="请输入用户名">
        </div>
    </div>
    <div class="form-group">
        <label for="password" class="col-sm-2 control-label">密码</label>
        <div class="col-sm-10">
            <input type="password" class="form-control" id="password"
                    placeholder="请输入密码">
        </div>
    </div>
    <div class="form-group">
        <div class="col-sm-offset-2 col-sm-10">
            <div class="checkbox">
                <label>
                    <input type="checkbox"> 请记住我
                </label>
            </div>
        </div>
    </div>
    <div class="form-group">
        <div class="col-sm-offset-2 col-sm-10">
            <button type="submit" class="btn btn-default">登录</button>
        </div>
    </div>
</form>
</body>
</html>
```

用浏览器打开 demo8-6，页面效果如图 8-22 所示。

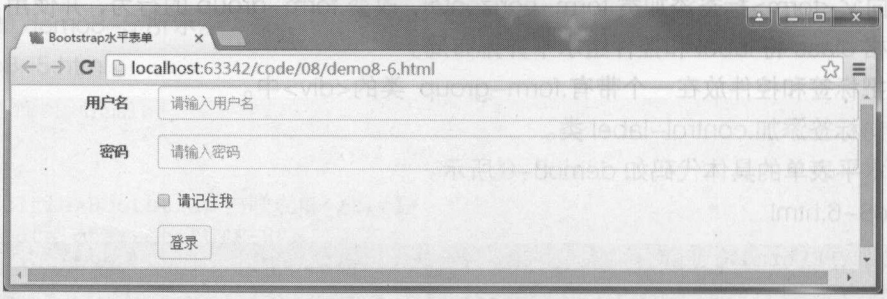


图8-22　demo8-6页面效果

4. 改变表单控件的样式

在实际开发中有时需要改变表单的默认尺寸和样式，可以通过如下方式来改变表单控件的尺寸和样式。

（1）使用.input-lg 和.input-sm 为控件设置高度。

（2）通过.col-lg-*为控件设置宽度。

（3）通过覆盖.form-control 的样式来改变控件的样式。

改变表单控件样式的具体代码如 demo8-7 所示。

demo8-7.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Bootstrap 改变表单控件的样式</title>
    <meta charset="UTF-8">
    <meta http-equiv="x-ua-compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="lib/bootstrap/css/bootstrap.min.css" rel="stylesheet">
</head>
<style>
    .form-control{
       background: pink;
    }
</style>
<body>
<form class="form-horizontal" role="form">
    <input class="form-control
          input-lg"type="text"placeholder=".input-lg">
    <input class="form-control" type="text"placeholder="Defaultinput">
    <div class="col-lg-3">
       <input class="form-control input-lg" type="text"
             placeholder= "input-lg col-lg-3"></p>
    </div>
    <input class="form-control input-sm" type="text"
          placeholder="input-sm"></p>
</form>
</body>
</html>
```

用浏览器打开 demo8-7，页面效果如图 8-23 所示。

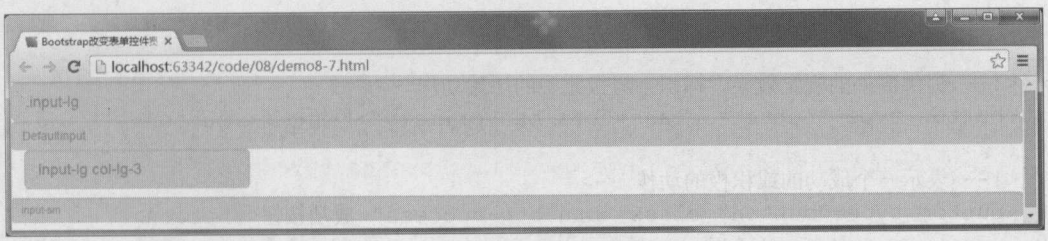


图8-23　demo8-7页面效果

在 demo8-7 中分别演示了使用不同类样式的<input>效果，并且覆盖.form-control 修改了

背景色样式，需要注意的是，类 col-lg-*需要在 标签上使用才能生效。

按钮

提到表单不得不说 Bootstrap 的按钮。使用 Bootstrap 定义按钮十分简单，可以为任何元素添加类.btn，按钮的默认外观为圆角灰色。

1. 按钮样式

Bootstrap 提供了一些类来定义按钮的样式，支持<a>、<button>和<input>标签，具体如表 8-1 所示。

表 8-1 Bootstrap 按钮选项

类	描 述
.btn	为按钮添加基本样式
.btn-default	默认/标准按钮
.btn-primary	原始按钮样式（未被操作）
.btn-success	表示成功的动作
.btn-info	该样式可用于要弹出信息的按钮
.btn-warning	表示需要谨慎操作的按钮
.btn-danger	表示一个危险动作的按钮操作
.btn-link	让按钮看起来像个链接（仍然保留按钮行为）
.active	按钮被单击
.disabled	禁用按钮

表 8-1 中的不同样式按钮的具体应用如 demo8-8 所示。

demo8-8.html

```
<!DOCTYPE html>
<html>
<head>
<title>Bootstrap 按钮</title>
<meta charset="UTF-8">
<meta http-equiv="x-ua-compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link href="lib/bootstrap/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
<!-- 标准的按钮 -->
<button type="button" class="btn btn-default">默认按钮</button>

<!-- 提供额外的视觉效果，标识一组按钮中的原始动作 -->
<button type="button" class="btn btn-primary">原始按钮</button>

<!-- 表示一个成功的或积极的动作 -->
<button type="button" class="btn btn-success">成功按钮</button>

<!-- 信息警告消息的上下文按钮 -->
<button type="button" class="btn btn-info">信息按钮</button>
```

```
22
23 <!-- 表示应谨慎采取的动作 -->
24 <button type="button" class="btn btn-warning">警告按钮</button>
25
26 <!-- 表示一个危险的或潜在的负面动作 -->
27 <button type="button" class="btn btn-danger">危险按钮</button>
28
29 <!-- 并不强调是一个按钮，看起来像一个链接，但同时保持按钮的行为 -->
30 <button type="button" class="btn btn-link">链接按钮</button>
31 </body>
32 </html>
33 </body>
34 </html>
```

用浏览器打开 demo8-8，页面效果如图 8-24 所示。

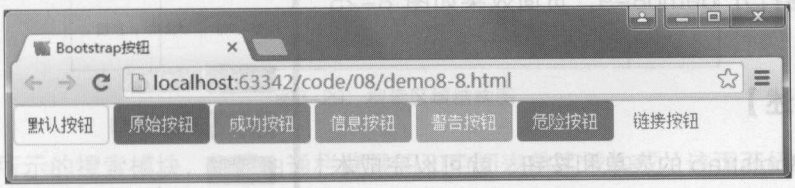


图8-24　demo8-8页面效果

2. 按钮大小

Bootstrap 中提供了一些类用于控制按钮的大小，如表 8-2 所示。

表 8-2　Bootstrap 按钮大小

类	描　述
.btn-lg	大按钮
.btn-sm	小按钮
.btn-xs	超小按钮
.btn-block	创建块级的按钮，会横跨父元素的全部宽度

表 8-2 中类的具体使用如 demo8-9 所示。

demo8-9.html

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <title>Bootstrap 按钮大小</title>
5  <meta charset="UTF-8">
6  <meta http-equiv="x-ua-compatible" content="IE=edge">
7  <meta name="viewport" content="width=device-width, initial-scale=1.0">
8  <link href="lib/bootstrap/css/bootstrap.min.css" rel="stylesheet">
9  </head>
10 <body style="padding: 20px;width:500px;">
11 <!-- 标准的按钮 -->
12 <button type="button" class="btn btn-default">默认按钮</button>
13 <button type="button" class="btn btn-lg btn-default">大的默认按钮</button>
```

```
<br/><br/>
<!-- 提供额外的视觉效果，标识一组按钮中的原始动作 -->
<button type="button" class="btn btn-primary">原始按钮</button>
<button type="button" class="btn btn-sm btn-primary">小的原始按钮</button>
<br/><br/>
<!-- 表示一个成功的或积极的动作 -->
<button type="button" class="btn btn-success">成功按钮</button>
<button type="button" class="btn btn-xs btn-success">特别小的成功按钮</button>
<br/><br/>
<!-- 用于要弹出信息的按钮 -->
<button type="button" class= "btn btn-info">信息按钮</button><br/> <br/>
<button type="button" class="btn btn-block btn-info">块级信息按钮</button>
</body>
</html>
```

用浏览器打开 demo8-9，页面效果如图 8-25 所示。

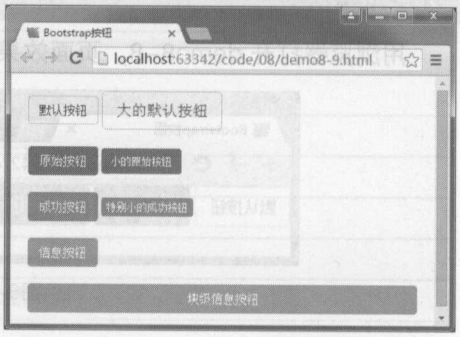


图8-25　demo8-9页面效果

【任务描述】

学习了 Bootstrap 的表单和按钮，就可以完成本单元项目的搜索区域了。该项目搜索区域的页面效果如图 8-26 所示。

浏览器窗口缩小至小屏幕（<992px）时，页面表单控件会随之发生变化，如图 8-27 所示。

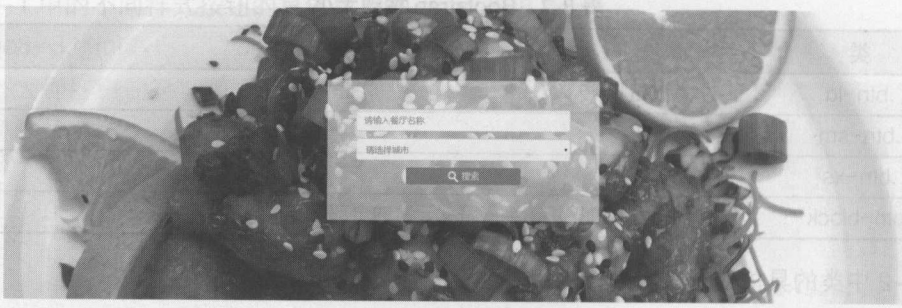


图8-26　搜索区域的页面效果

当浏览器窗口缩小至超小屏幕（<768px）时，保证页面表单依然完整，如图 8-28 所示。

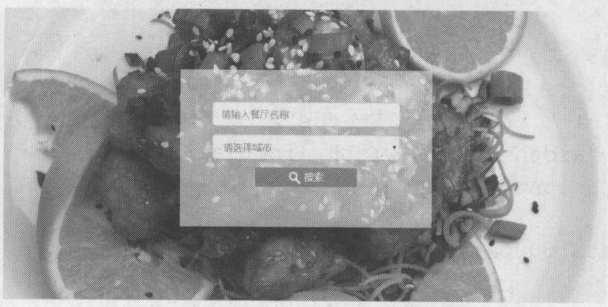


图8-27　小屏幕页面效果

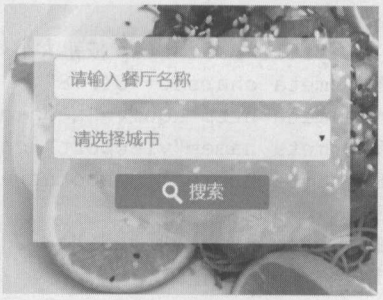


图8-28　超小屏幕页面效果

【任务分析】

了解了该任务要实现的效果后，接下来我们分析一下页面结构，如图 8-29 所示。

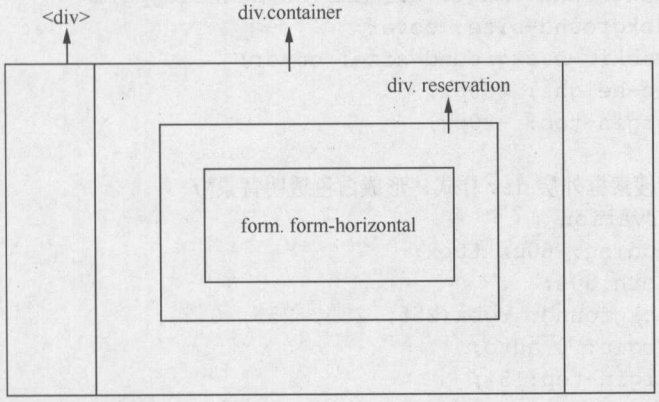


图8-29　搜索区域结构图

图 8-29 所示的搜索模块，主要由通栏背景图和中间表单区域构成。该页面的实现细节具体分析如下。

（1）在最外层添加一个 标签，并且设置通栏的背景图片。

（2）使用布局容器带有.container 类的 标签进行布局，在该 中添加一个 ，为该 标签设置宽度和较大的 padding 值来实现表单的白色透明背景。

（3）白色背景 div.reservation 中添加 Bootstrap 水平表单，并且删掉表单中没有用到的<label>等标签。

（4）通过.btn 类来修改 Bootstrap 按钮的默认样式。

（5）使用媒体查询，在屏幕小于 768px 时，让表单外层的 在小屏幕时拥有足够的宽度来显示完整的表单，并且设置 body、button、 input、 select 元素的字体为微软雅黑。

【代码实现】

对页面的结构有所了解后，即可开始编写代码来实现它。该任务的代码如 code\08\0801\0801_2.html 所示。

0801_2.html

```
<!DOCTYPE html>
<html>
<head>
    <title>餐饮网站搜索区域</title>
    <meta charset="UTF-8">
    <meta http-equiv="x-ua-compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- 引入 Bootstrap -->
    <link href="lib/bootstrap/css/bootstrap.min.css" rel="stylesheet">
    <style type="text/css">
        /*为页面中以下元素设置字体为 Microsoft YaHei*/
```

```
        body,button, input, select { font-family: Microsoft YaHei}
        /*搜索区域*/
        /*设置“搜索”按钮前面的搜索图标*/
        .search{
            background-image: url(images/banner.jpg);
            background-size: cover;
            -webkit-background-size: cover;
            min-height: 600px;
            margin-top: -20px;
        }
        /*设置搜索框外层 div 样式，形成白色透明背景*/
        .reservation {
            padding: 60px 60px;
            width:50%;
            background: rgba(255, 255, 255, 0.7);
            margin: 0 auto;
            margin-top:15%;
            font-weight: 500;
            color: #f2f0f1;
            font-size: 1.2em;
            outline: none;
        }
        /*设置“搜索”按钮的样式*/
        .btn{
            width: 50% ;
            background: #D96B66;
            color: #fff;
            padding: 5px;
            border: none;
            border-radius: 4px;
            -webkit-border-radius: 4px;
        }
        .form-control{
            color: #8e908d;
        }
        .searchbtn{
            text-align: center;
        }
        /*媒体查询：当屏幕小于 768px 时搜索表单白色背景大小发生变化*/
        @media (max-width: 768px){
            .reservation {
                padding: 20px 20px;
                width: 90%;
            }
        }
    </style>
</head>
<body>
<!--搜索区域-->
<div class="search" >
```

```
    <div class="container">
        <div class="reservation">
            <form class="form-horizontal" role="form">
                <div class="form-group">
                    <div class="col-sm-12 col-md-12 col-lg-12">
                        <input type="text" class="form-control input-lg"
                               id="name" placeholder="请输入餐厅名称">
                    </div>
                </div>
                <div class="form-group">
                    <div class="col-sm-12 col-md-12 col-lg-12">
                        <select id="country"  class="form-control input-lg">
                            <option value="null">请选择城市</option>
                            <option value="bj">北京</option>
                            <option value="sh">上海</option>
                            <option value="sz">深圳</option>
                        </select>
                    </div>
                </div>
                <div class="form-group">
                    <div class="searchbtn">
                        <button type="submit" class="btn btn-success btn-lg">
                         <img src="images/search-icon.png"> 搜索</button>
                    </div>
                </div>
            </form>
        </div>
    </div>
</div>
<!-- 搜索区域结束 -->
<!-- jQuery (Bootstrap 的 JavaScript 插件需要引入 jQuery) -->
<script src="lib/jquery/jquery-1.11.0.min.js"></script>
<!-- 包括所有已编译的插件 -->
<script src="lib/bootstrap/js/bootstrap.min.js"></script>
</body>
</html>
```

【任务 3-完成热卖商品模块】

【前导知识】

栅格系统

Bootstrap 提供了一套响应式、移动设备优先的流式栅格系统，随着屏幕或视口（viewport）尺寸的增加，系统会自动分为最多 12 列。它包含了易于使用的预定义类和强大的 mixin 用于生成更具语义的布局。

栅格系统用于通过一系列的行（row）与列（column）的组合来创建页面布局，开发者可以将内容放入这些创建好的布局中。

Bootstrap 栅格系统的工作原理如下。

- “行”必须包含在布局容器.container 类或.container-fluid 类中，以便为其赋予合适的排列（aligment）和内补（padding）。
- 通过“行（row）”在水平方向创建一组“列（column）”，并且只有“列（column）”可以作为“行（row）”的直接子元素。
- 行使用样式“.row”，列使用样式“col-*-*”，我们的内容应当放置于“列（column）”内，列大于 12 时，将另起一行排列。
- Bootstrap 栅格系统为不同屏幕宽度定义了不同的类。

Bootstrap3 使用了 4 种栅格选项来形成栅格系统，这 4 种栅格选项的区别在于适合不同尺寸的屏幕设备，官网上的具体介绍如图 8-30 所示。

	超小屏幕 手机 (<768px)	小屏幕 平板 (≥768px)	中等屏幕 桌面显示器 (≥992px)	大屏幕 大桌面显示器 (≥1200px)
栅格系统行为	总是水平排列	开始是堆叠在一起的，当大于这些阈值时将变为水平排列C		
.container 最大宽度	None（自动）	750px	970px	1170px
类前缀	.col-xs-	.col-sm-	.col-md-	.col-lg-
列（column）数	12			
最大列（column）宽	自动	~62px	~81px	~97px
槽（gutter）宽	30px（每列左右均有 15px）			
可嵌套	是			
偏移（Offsets）	是			
列排序	是			

图8-30　栅格参数

在图 8-30 中，“类前缀”这一项的取值分别是 col-xs、col-sm、col-md、col-lg，其中，lg 是 large（大）的缩写，md 是 mid（中等）的缩写，sm 是 small（小）的缩写，xs 是 extrasmall（超小）的缩写。这样命名就体现了这几种 class 适应不同的屏幕宽度，如使用.col-xs 适合于小于 768px 的超小屏幕。

栅格系统可以让用户在不同尺寸的设备上看见不同的布局效果，具体应用如 demo8-10 所示。

demo8-10.html

```
<!DOCTYPE html>
<html>
<head>
    <title>栅格系统</title>
    <meta charset="UTF-8">
    <meta http-equiv="x-ua-compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="lib/bootstrap/css/bootstrap.min.css" rel="stylesheet">
</head>
<style>
    div{
        border: 1.5px solid #000000;
    }
</style>
```

```
<br>
<div class="container">
    <div class="row">
        <div class="col-md-3 col-xs-6">1</div>
        <div class="col-md-3 col-xs-6">2</div>
        <div class="col-md-3 col-xs-6">3</div>
        <div class="col-md-3 col-xs-6">4</div>
    </div>
    <div class="row">
        <div class="col-md-3 col-xs-6">5</div>
        <div class="col-md-3 col-xs-6">6</div>
        <div class="col-md-3 col-xs-6">7</div>
        <div class="col-md-3 col-xs-6">8</div>
    </div>
</div>
<script src="lib/jquery/jquery-1.11.0.min.js"></script>
<script src="lib/bootstrap/js/bootstrap.min.js"></script>
</body>
</html>
```

用浏览器打开 demo8-10，页面效果如图 8-31 所示。

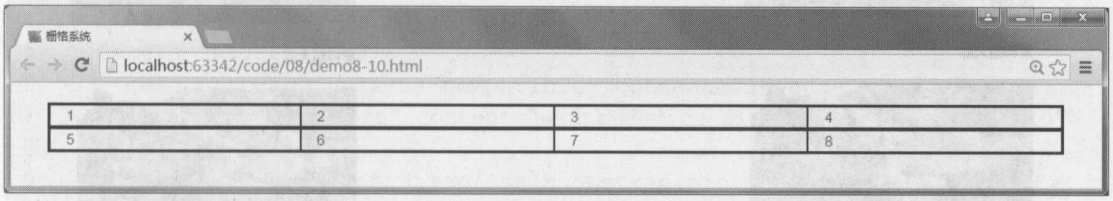


图8-31　demo8-10页面效果

使用鼠标拖动，缩小浏览器窗口至小屏幕，页面网格会变成两列，如图 8-32 所示。

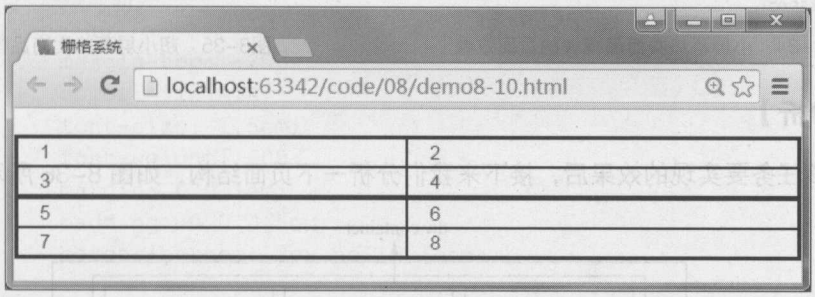


图8-32　缩小浏览器的效果

图 8-32 所示的效果是由于浏览器窗口缩小至小于 768px，会被识别为手机设备大小，.col-xs-6 设置生效，当列大于 12 时另起一行排列，所以显示了 4 行。

【任务描述】

学习了 Bootstrap 的栅格系统，就可以完成本单元项目的热卖商品模块了。该项目热卖商品模块的页面效果如图 8-33 所示。

图8-33　热卖商品模块的页面效果

当浏览器缩小至小屏幕（<992px）时，热卖商品每行显示一个，页面效果如图 8-34 所示。当浏览器缩小至超小屏幕（<768px）时，热卖商品依然每行显示一个，如图 8-35 所示。

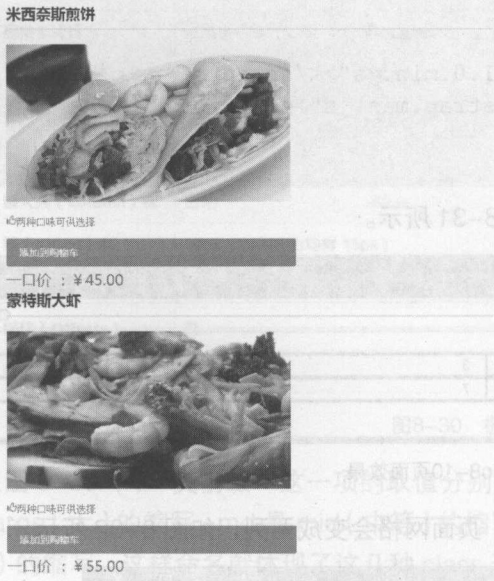


图8-34　小屏幕热卖商品模块的页面效果

图8-35　超小屏幕热卖商品模块的页面效果

【任务分析】

了解了该任务要实现的效果后，接下来我们分析一下页面结构，如图 8-36 所示。

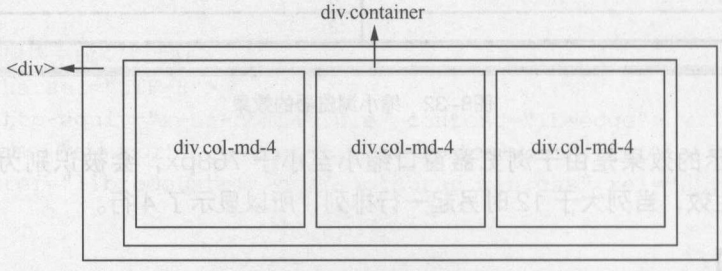


图8-36　热卖商品模块结构图

图 8-36 所示的热卖商品模块，由最外层 标签嵌套多个小 标签构成。该页面的实现细节具体分析如下。

（1）最外层 标签自定义类名.hot，用于批量设置其子元素的样式。

（2）在最外层 下添加 标签并使用类.container 实现整体布局，在 div.container 添加栅格系统，3 个 分别使用类名.col-md-4，分别在 3 个 中添加商品的图片和商品信息，Bootstrap 会在小屏幕时自动变为每行一个商品。

（3）为了使按钮宽度和图片宽度一致，这里使用的是自定义的按钮。另外，商品信息中的“👍”图标使用了 Bootstrap 的字体图标（glyphicon glyphicon-thumbs-up），Bootstrap 提供的字体图标和之前介绍过的字体图标使用方式相似，只需要添加相应的类名即可，具体图标对应类名可以查看相关手册。

（4）设置 body、h3、h6 元素的字体为微软雅黑。

【代码实现】

对页面的结构有所了解后，即可开始编写代码来实现它。该任务的代码如 code\08\0801\0801_3.html 所示。

0801_3.html

```
<!DOCTYPE html>
<html>
<head>
    <title>餐饮网站热卖商品模块</title>
    <meta charset="UTF-8">
    <meta http-equiv="x-ua-compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- 引入 Bootstrap -->
    <link href="lib/bootstrap/css/bootstrap.min.css" rel="stylesheet">
    <style type="text/css">
        /*为页面中以下元素设置字体为 Microsoft YaHei*/
        body,h3,h6 { font-family: Microsoft YaHei}
        /*热卖商品*/
        .hot h3 {
            margin-top: -20px;
            color: #1A1A1A;
            font-size: 1.5em;
            font-weight: 600;
            margin: 0 0 1em;
            padding: 0 0 0.5em;
            border-bottom: 2px solid #eee;
        }
        .hot p {
            color: #5fa022;;
            font-size: 1em;
            font-weight: 400;
            line-height: 1.8em;
            margin: 1em 0;
        }
        .hot {
            padding: 3em 0;
            border: 1px solid #eee;
```

```
33          margin: 0 0;
34      }
35      .hot h6 {
36          color: #C15162;
37          font-size: 1.5em;
38          font-weight: 400;
39          margin: 0.3em 0;
40      }
41      a.morebtn {
42          display: block;
43          font-size: 1em;
44          color: #FFF;
45          text-decoration: none;/*去掉 a 链接的下划线*/
46          font-weight: 400;
47          background: #D96B66;
48          padding: 10px 18px;
49          transition: 0.5s all ease;
50          -webkit-transition: 0.5s all ease;
51          border-radius: 3px;
52          -webkit-border-radius: 3px;
53      }
54      a.morebtn:hover{
55          background:  #5fa022;
56      }
57      @media (max-width: 1024px) {
58          a.morebtn{
59              max-width:410px;/*当视口小于 1024px 时按钮的最大宽度*/
60          }
61      }
62  </style>
63 </head>
64 <body>
65 <!-- 热卖商品 -->
66 <div class="hot">
67     <div class="container">
68         <div class="col-md-4 ">
69             <h3>米西奈斯煎饼</h3>
70             <img src="images/4.jpg" class="img-responsive" alt="">
71             <div>
72                 <p class="glyphicon glyphicon-thumbs-up">两种口味可供选择 </p>
73                 <div class="cur">
74                     <span ><a class="morebtn" href="#">添加到购物车</a></span>
75                     <span><h6>一口价 ： ￥45.00</h6></span>
76                 </div>
77             </div>
78         </div>
79         <div class="col-md-4">
80             <h3>蒙特斯大虾</h3>
81             <img src="images/1.jpg" class="img-responsive" alt="蒙特斯大虾">
82             <div>
```

```
                <p class="glyphicon glyphicon-thumbs-up">两种口味可供选择 </p>
                <div class="cur">
                    <span><a class="morebtn" href="#">添加到购物车</a></span>
                    <span><h6>一口价 ：￥55.00</h6></span>
                </div>
            </div>
        </div>
        <div class="col-md-4">
            <h3>香酥鸡排</h3>
            <img src="images/3.jpg" class="img-responsive" alt="香酥鸡排">
            <div>
                <p class="glyphicon glyphicon-thumbs-up">两种口味可供选择 </p>
                <div class="cur">
                    <span><a class="morebtn " href="#">添加到购物车</a></span>
                    <span><h6>一口价 ：￥65.00</h6></span>
                </div>
            </div>
        </div>
    </div>
</div>
<!--热卖商品结束 -->
<!-- jQuery (Bootstrap 的 JavaScript 插件需要引入 jQuery) -->
<script src="lib/jquery/jquery-1.11.0.min.js"></script>
<!-- 包括所有已编译的插件 -->
<script src="lib/bootstrap/js/bootstrap.min.js"></script>
</body>
</html>
```

【任务 4-完成特色推荐模块】

【前导知识】

标签页

Bootstrap 提供了几种标签页，这里我们讲解一下应用比较广泛的胶囊标签页。Bootstrap 制作胶囊标签页主要分为以下步骤。

（1）使用一个完整的标签页分为页头选项卡和内容两部分。

（2）页头使用<ul>标签，在<ul>中添加.nav 和.nav-tabs 类，会应用 Bootstrap 标签页样式；添加.nav 和. nav-pills 类会应用胶囊标签样式。需要几个标签项就添加几个<li>标签。

（3）在<li>标签中添加<a>标签，<a>标签的 href 的值直接与标签页下面的内容 的 id 关联，十分重要。

（4）在<a>标签中添加 data-toggle=“tab”或 data-toggle=“pill”。页头部分示例代码如下所示。

```
<ul class="nav nav-tabs">
   <li><a href="#identifier" data-toggle="tab">Home</a></li>
...
</ul>
```

（5）内容部分最外层使用 标签添加类.tab-content，然后添加每个标签项对应的 标签，这些标签添加类.tab-pane 和对应标签项的 id 值，示例代码如下所示。

```
<div class="tab-content">
<div class="tab-pane active" id="home">...</div>
<div class="tab-pane " id="profile">...</div>
<div class="tab-pane " id="messages">...</div>
</div>
```

在上述代码中，.active 类用来定义当前选中项。接下来，我们通过一个案例演示最基本的胶囊标签页，具体代码如 demo8-11 所示。

demo8-11.html

```
<!DOCTYPE html>
<html>
<head>
<title>胶囊标签页</title>
<meta charset="UTF-8">
<meta http-equiv="x-ua-compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link href="lib/bootstrap/css/bootstrap.min.css" rel="stylesheet">
</head>
<br>
<ul class="nav nav-pills">
<li role="presentation" class="active"><a href="#home"
                    role="tab" data- toggle="pill">Home</a></li>
<li role="presentation"><a href="#profile" role="tab"
                    data-toggle="pill"> Profile</a></li>
<li role="presentation"><a href="#messages" role="tab"
                    data-toggle="pill"> Messages</a></li>
</ul>
<div class="tab-content">
<div class="tab-pane  active" id="home">我是第一页</div>
<div class="tab-pane " id="profile">我是第二页</div>
<div class="tab-pane " id="messages">我是第三页</div>
</div>
<script src="lib/jquery/jquery-1.11.0.min.js"></script>
<script src="lib/bootstrap/js/bootstrap.min.js"></script>
</body>
</html>
```

用浏览器打开 demo8-11，页面效果如图 8-37 所示。

单击“Messages”选项，会切换到第三页，如图 8-38 所示。

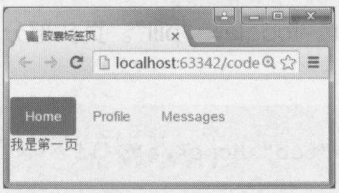


图8-37　demo8-11页面效果

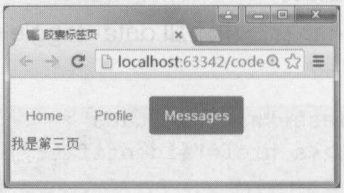


图8-38　demo8-11选项切换效果

响应式工具

为了更快地实现对移动设备的友好开发，Bootstrap 提供了一套辅助工具类，使用这些工具类可以通过媒体查询结合大型、小型和中型设备，实现内容在设备上的显示和隐藏。

目前，可供使用的类如表 8-3 所示。

表 8-3　Bootstrap 响应式工具类

类名	超小屏幕 手机（<768px）	小屏幕 平板（≥768px）	中等屏幕 桌面（≥992px）	大屏幕 桌（≥1200px）
.visible-xs-*	可见	隐藏	隐藏	隐藏
.visible-sm-*	隐藏	可见	隐藏	隐藏
.visible-md-*	隐藏	隐藏	可见	隐藏
.visible-lg-*	隐藏	隐藏	隐藏	可见
.hidden-xs	隐藏	可见	可见	可见
.hidden-sm	可见	隐藏	可见	可见
.hidden-md	可见	可见	隐藏	可见
.hidden-lg	可见	可见	可见	隐藏

表 8-3 中的响应式实用工具目前只适用于块级元素和表格的切换，具体使用如 demo8-12 所示。

demo8-12 .html

```
<!DOCTYPE html>
<html>
<head>
<title>响应式工具</title>
<meta charset="UTF-8">
<meta http-equiv="x-ua-compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link href="lib/bootstrap/css/bootstrap.min.css" rel="stylesheet">
<style>
        div{
            border: 1px solid black;
        }
</style>
</head>
<body>
<br>
<div class="container" style="padding: 40px;">
<div class="row visible-on">
<div class="col-xs-6 col-sm-3" >
<span class="hidden-xs">特别小型设备隐藏</span>
<span class="visible-xs">✔在特别小型设备上可见</span>
</div>
<div class="col-xs-6 col-sm-3" >
<span class="hidden-sm">小型设备隐藏</span>
```

```
25 <span class="visible-sm">✔在小型设备上可见</span>
26 </div>
27 <div class="clearfix visible-xs"></div>
28 <div class="col-xs-6 col-sm-3" >
29 <span class="hidden-md">中型设备隐藏</span>
30 <span class="visible-md">✔在中型设备上可见</span>
31 </div>
32 <div class="col-xs-6 col-sm-3" >
33 <span class="hidden-lg">大型设备隐藏</span>
34 <span class="visible-lg">✔在大型设备上可见</span>
35 </div>
36 </div>
37 </div>
38 </body>
39 </html>
```

用浏览器打开 demo8-12，页面效果如图 8-39 所示。

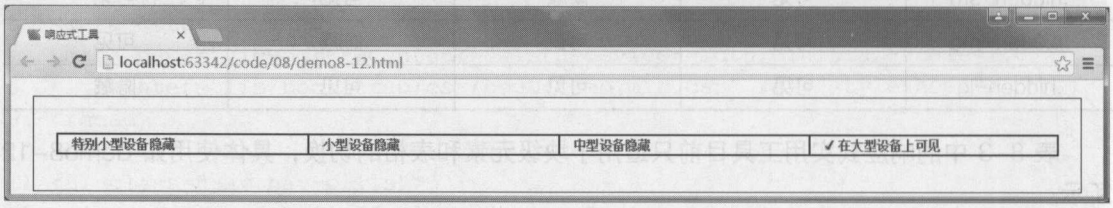


图8-39　demo8-12页面效果

浏览器窗口缩小至中型屏幕时，页面效果如图 8-40 所示。

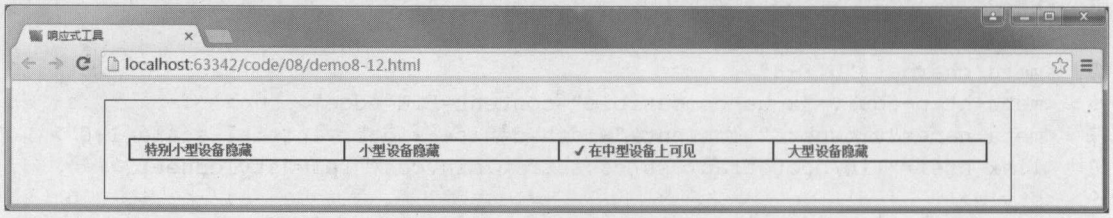


图8-40　中型屏幕页面效果

浏览器窗口缩小至小型屏幕时，页面效果如图 8-41 所示。

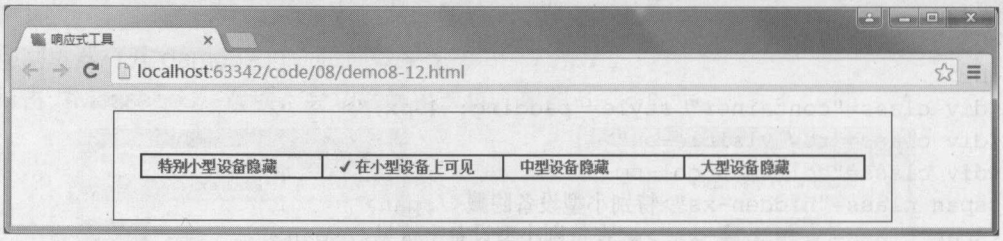


图8-41　小型屏幕页面效果

浏览器窗口缩小至超小型屏幕时，页面效果如图 8-42 所示。

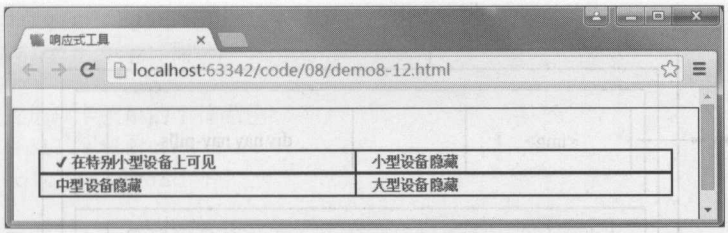


图8-42　超小型屏幕页面效果

【任务描述】

学习了 Bootstrap 的标签页和响应式工具后，就可以完成本单元项目的特色推荐模块了。该项目中该模块的页面效果如图 8-43 所示。

图8-43　特色推荐模块页面效果

浏览器窗口缩小至小屏幕（<992px）时，推荐商品每行显示 3 个，页面效果如图 8-44 所示。

当浏览器缩窗口小至超小屏幕（<768px）时，推荐商品变为每行两个，并且隐藏左上角图片，选项卡变为左侧显示，如图 8-45 所示。

图8-44　小屏幕特色推荐模块页面效果

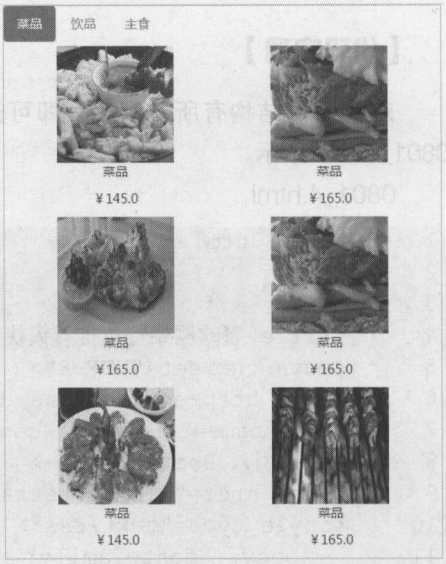


图8-45　超小屏幕热卖商品

【任务分析】

了解了该任务要实现的效果后，接下来我们分析一下页面结构，如图 8-46 所示。

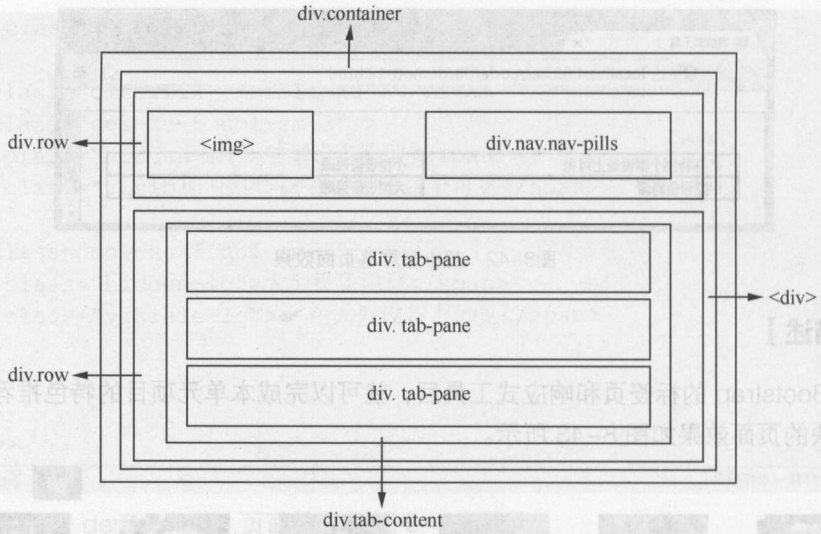


图8-46　特色推荐结构图

图 8-46 所示的特色推荐模块主要通过 Bootstrap 标签页来实现。该页面的实现细节具体分析如下。

（1）首先应在最外层添加一个使用.container 类的 标签，在该标签中添加自定义类名.choose 的 标签，并且添加灰色边框。

（2）在 div.choose 中添加两个使用.row 类的 标签，第一个用来添加选项卡，第二个用来添加选项内容，左上角图片位置与导航栏的 logo 类似，可以嵌套在使用.nav-brand 类的<a>标签中。

（3）应用栅格系统来实现每页商品的响应式布局，小屏幕时隐藏图片效果使用响应式工具“hidden-xs”即可。

【代码实现】

对页面的结构有所了解后，即可开始编写代码来实现它。该任务的代码如 code\08\0801\0801_4.html 所示。

0801_4.html

```
<!DOCTYPE html>
<html>
<head>
    <title>餐饮网站特色推荐模块</title>
    <meta charset="UTF-8">
    <meta http-equiv="x-ua-compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- 引入 Bootstrap -->
    <link href="lib/bootstrap/css/bootstrap.min.css" rel="stylesheet">
    <style type="text/css">
        body { font-family: Microsoft YaHei}/*设置 body 元素字体*/
        /*特色推荐*/
        /*设置左上角图片的位置和宽高*/
        .navbar-brand>img{
            height: auto;
            width:250px;
```

```
17          margin-right: 5px;
18          margin-top: 5px;
19       }
20       /*设置选项卡菜单的字体颜色*/
21       .nav-pills>li>a{
22          color:#8e908d ;
23       }
24       /*设置选项卡菜单鼠标悬停和获取焦点时的背景色和字体颜色*/
25       .nav-pills>li.active>a, .nav-pills>li.active>a:focus,
26       .nav-pills>li.active>a:hover {
27          color: #fff;
28          background-color: #5A9522;
29       }
30       .choose{
31          border: 1px solid silver;
32       }
33       /*设置选项卡内容的位置*/
34       .tab-content{
35          margin: 5px;
36          text-align: center;/*居中显示*/
37       }
38    </style>
39 </head>
40 <body>
41 <!--特色推荐-->
42 <div class="container">
43    <div class="choose">
44       <div class="row">
45          <div class="col-md-12">
46             <div class="navbar-header hidden-xs">
47                <a class="navbar-brand" href="#"><img
                   src="images/title. jpg" sizes="100px"/></a>
48             </div>
49             <!-- 选项卡菜单 -->
50             <ul class="nav nav-pills navbar-right " role="tablist"
                 style= "margin-right: 0px;">
51                <li role="presentation" class="active"><a href="#dishes"
                   role="tab" data-toggle="tab">菜品</a></li>
52                <li role="presentation"><a href="#drink"  role="tab"
                   data-toggle="tab">饮品</a></li>
53                <li role="presentation"><a href="#staple"  role="tab"
                  data-toggle="tab">主食</a></li>
54             </ul>
55          </div>
56       </div>
57       <div class="row">
58          <div class="col-md-12">
59             <div class="tab-content">
60                <!--菜品-->
61                <div role="tabpanel" class="tab-pane active" id="dishes">
```

```
62                  <div class=" col-md-2 col-sm-4 col-xs-6">
63                      <img src="images/products/small01.jpg"/>
64                      <p>菜品</p>
65                      <p>￥145.0</p>
66                  </div>
67                  <div class="product-item col-md-2 col-sm-4 col-xs-6">
68                      <img src="images/products/small02.jpg"/>
69                      <p>菜品</p>
70                      <p>￥165.0</p>
71                  </div>
72                  <div class="product-item col-md-2 col-sm-4 col-xs-6">
73                      <img src="images/products/small04.jpg"/>
74                      <p>菜品</p>
75                      <p>￥165.0</p>
76                  </div>
77                  <div class="product-item col-md-2 col-sm-4 col-xs-6">
78                      <img src="images/products/small02.jpg"/>
79                      <p>菜品</p>
80                      <p>￥165.0</p>
81                  </div>
82                  <div class="product-item col-md-2 col-sm-4 col-xs-6">
83                      <img src="images/products/small07.jpg"/>
84                      <p>菜品</p>
85                      <p>￥145.0</p>
86                  </div>
87                  <div class="product-item col-md-2 col-sm-4 col-xs-6">
88                      <img src="images/products/small08.jpg"/>
89                      <p>菜品</p>
90                      <p>￥165.0</p>
91                  </div>
92              </div>
93              <!--饮品-->
94              <div role="tabpanel" class="tab-pane" id="drink">
95                  <div class="product-item col-md-2 col-sm-4 col-xs-6">
96                      <img src="images/products/small03.jpg"/>
97                      <p>饮品</p>
98                      <p>￥98.0</p>
99                  </div>
100                 <div class="product-item col-md-2 col-sm-4 col-xs-6">
101                     <img src="images/products/small03.jpg"/>
102                     <p>饮品</p>
103                     <p>￥98.0</p>
104                 </div>
105             </div>
106             <!--</div>-->
107             <!--主食-->
108             <div role="tabpanel" class="tab-pane" id="staple">
109                 <div class="product-item col-md-2 col-sm-4 col-xs-6">
110                     <img src="images/products/small05.jpg"/>
111                     <p>主食</p>
```

```
                        <p>¥56.0</p>
                    </div>
                    <div class="product-item col-md-2 col-sm-4 col-xs-6">
                        <img src="images/products/small06.jpg"/>
                        <p>主食</p>
                        <p>¥56.0</p>
                    </div>
                    <div class="product-item col-md-2 col-sm-4 col-xs-6">
                        <img src="images/products/small09.jpg"/>
                        <p>主食</p>
                        <p>¥56.0</p>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
<!--特色推荐结束-->
<!-- jQuery (Bootstrap 的 JavaScript 插件需要引入 jQuery) -->
<script src="lib/jquery/jquery-1.11.0.min.js"></script>
<!-- 包括所有已编译的插件 -->
<script src="lib/bootstrap/js/bootstrap.min.js"></script>
</body>
</html>
```

【任务 5-完成轮播广告模块】

【前导知识】

轮播插件（Carousel）

Bootstrap 轮播插件（Carousel）是一种灵活的、响应式的、向站点添加滑块的方式。轮播的内容可以是图像、内嵌框架、视频或者其他想要放置的任何类型的内容，使用该插件时必须引入 bootstrap.js 或压缩版的 bootstrap.min.js。

接下来，我们以轮播图片为例讲解 Bootstrap 轮播插件的使用，轮播图的实现主要由 3 个部分构成：轮播的图片、轮播图片的计数器和轮播图片的控制器。

（1）设计轮播容器

使用.carousel 类设计轮播图片的容器，并为该容器添加 id，方便后面的使用，示例代码如下所示。

```
<div id="slidershow" class="carousel">
...
</div>
```

（2）设计轮播计数器

在轮播容器 div.carousel 的内部添加轮播计算器.carousel-indicators 类，其主要功能是显示当前图片的播放顺序（有几张图片就放置几个 li），一般采用有序列表来制作。该内容放在轮

播容器内，示例代码如下所示。

```
<!-- 设置图片轮播的顺序 -->
 <ol class="carousel-indicators">
    <li class="active">1</li>
    <li>2</li>
    <li>3</li>
    <li>4</li>
    <li>5</li><
</ol>
```

（3）设计轮播图片控制器

很多时候，轮播图片还具有一个向前播放和向后播放的控制器。在 Carousel 中通过.carousel-control 类配合 left 和 right 来实现。其中，left 表示向前播放，right 表示向后播放。该内容同样放在 carousel 轮播容器内，示例代码如下所示。

```
<!-- 设置轮播图片控制器 -->
 <a class="left carousel-control" href="" >
    <span class="glyphicon glyphicon-chevron-left"></span>
 </a>
 <a class="right carousel-control" href="">
    <span class="glyphicon glyphicon-chevron-right"></span>
 </a>
```

（4）添加图片描述

Bootstrap 中可以使用 标签添加.carousel-caption 类为图片添加描述信息，这部分内容只需要在 div.item 中图片底部添加对应的代码，示例代码如下所示。

```
<!-- 图片对应标题和描述内容 -->
 <div class="carousel-caption">
    <h3>图片标题</h3>
    <p>描述内容...</p>
 </div>
```

（5）声明式触发轮播

声明式方法是通过定义 data 属性来实现的。通过 data 属性可以很容易地控制轮播的位置。data 属性主要包括以下几种。

- data-ride 属性：取值 carousel，并且将其定义在 carousel 上。
- data-target 属性：取值 carousel 定义的 ID 名或者其他样式识别符，如前面示例所示，取值为“#slidershow”，并且将其定义在轮播图计数器的每个<li>标签上。
- data-slide 属性：取值有 2 个，prev 表示向后滚动，next 表示向前滚动。该属性值同样定义在轮播图控制器的 a 链接上，同时设置控制器 href 值为容器 carousel 的 id 名或其他样式识别符。
- data-slide-to 属性：用来传递某个帧的下标，如 data-slide-to=“2”，可以直接跳转到这个指定的帧（下标从 0 开始计），同样定义在轮播图计数器的每个<li>标签上。

完整的图片轮播代码如 demo8-13 所示。

demo8-13.html

```
1  <!DOCTYPE html>
2  <html>
```

```
<head>
    <title>Bootstrap 轮播插件</title>
    <meta charset="UTF-8">
    <meta http-equiv="x-ua-compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="lib/bootstrap/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
<div id="slidershow" class="carousel slide" data-ride="carousel">
    <!-- 设置图片轮播计数器 -->
    <ol class="carousel-indicators">
        <li class="active" data-target="#slidershow" data-slide-to="0"></li>
        <li data-target="#slidershow" data-slide-to="1"></li>
        <li data-target="#slidershow" data-slide-to="2"></li>
    </ol>
    <!-- 设置轮播图片 -->
    <div class="carousel-inner">
        <div class="item active">
            <a href="#">
                <img src="images/demo8-13/carousel1.jpg"
                 style=" height: 400px;margin:0 auto;">
            </a>
            <div class="carousel-caption">
                <h3>图片标题 1</h3>
                <p>描述内容 1...</p>
            </div>
        </div>
        <div class="item">
            <a href="#">
                <img src="images/demo8-13/carousel2.jpg"
                 style="height:400px; margin:0 auto;">
            </a>
            <div class="carousel-caption">
                <h3>图片标题 2</h3>
                <p>描述内容 2...</p>
            </div>
        </div>
        <div class="item">
            <a href="#">
                 <img src="images/demo8-13/carousel3.jpg" style="
                 height: 400px;margin:0 auto;">
            </a>
            <div class="carousel-caption">
                <h3>图片标题 3</h3>
                <p>描述内容 3...</p>
            </div>
        </div>
    </div>
    <!-- 设置轮播图片控制器 -->
    <a class="left carousel-control" href="#slidershow" role="button"
       data-slide="prev">
        <span class="glyphicon glyphicon-chevron-left"></span>
```

```
52      </a>
53      <a class="right carousel-control" href="#slidershow" role="button"
54        data-slide="next">
55          <span class="glyphicon glyphicon-chevron-right"></span>
56      </a>
57  </div>
58  <footer class="footer navbar-fixed-bottom ">
59      <div class="container">
60      </div>
61  </footer>
62  <script src="lib/jquery/jquery-1.11.0.min.js"></script>
63  <script src="lib/bootstrap/js/bootstrap.min.js"></script>
64  </body>
65  </html>
```

用浏览器打开 demo8-13，页面效果如图 8-47 所示。

图8-47　demo8-13页面效果

图 8-47 所示的轮播图会在默认的时间后自动轮播，也可以单击左右轮播导航或者描述内容下方的圆点导航进行图片切换，效果如图 8-48 所示。

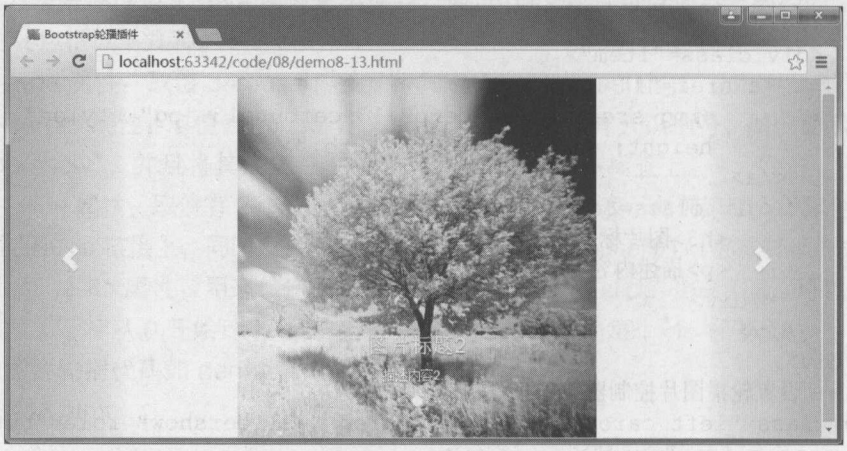


图8-48　轮播图切换页面效果

需要注意的是，这里使用了自定义样式设置了图片的高度和位置，第 2 张图片和第 1 张的宽度不同是因为应用了图片本身的宽度，在实际开发中轮播内容的样式可以根据需求进行自定义。

【任务描述】

学习了 Bootstrap 的轮播，就可以完成本单元项目的轮播广告模块了。该项目的轮播广告模块的页面效果如图 8-49 所示。

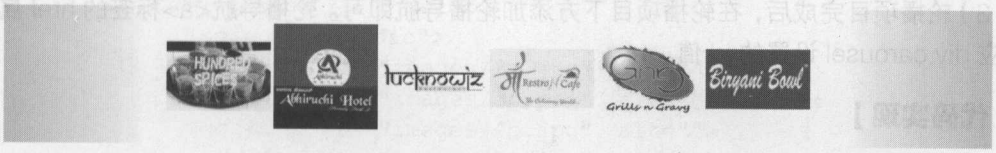


图8-49　轮播广告模块页面效果

当浏览器窗口缩小至小屏幕（<992px）时，轮播图片每行显示 3 个，页面效果如图 8-50 所示。

图8-50　小屏幕轮播广告

当浏览器窗口缩小至超小屏幕（<768px）时，轮播广告被隐藏。

【任务分析】

了解了该任务要实现的效果后，接下来我们分析一下页面结构，如图 8-51 所示。

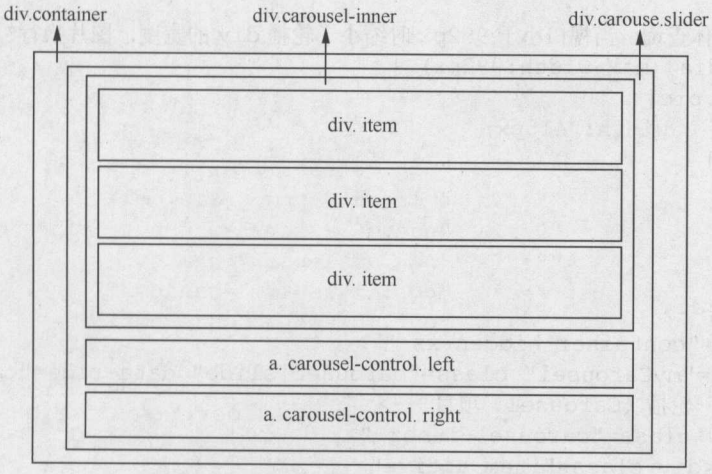


图8-51　轮播广告结构图

图 8-51 所示的轮播广告，主要使用 Bootstrap 的轮播插件来完成，该页面的实现细节具体分析如下。

（1）首先在最外层添加一个使用.container 类的 标签，在该标签内添加轮播 标签，使用.carousel 和.slider 类，并且设置自定义 id 值和 data-ride=“carousel”自动播放。

（2）在 div.item 中添加轮播项目，每个轮播项目放在使用.item 类的 标签中。需要注意的是，每个轮播项目要轮播 6 张图片，为了让这 6 张图片横向排列，需要在外层添加自定义 标签，并且为该标签设置宽度和居中显示，小屏幕时使用媒体查询重新设置宽度。

（3）轮播项目完成后，在轮播项目下方添加轮播导航即可。轮播导航<a>标签的 href 属性必须对应 div.carousel 设置的 id 值。

【代码实现】

对页面的结构有所了解后，即可开始编写代码来实现它。该任务的代码如 code\08\0801\0801_5.html 所示。

0801_5.html

```
<!DOCTYPE html>
<html>
<head>
<title>餐饮网站轮播广告模块</title>
<meta charset="UTF-8">
<meta http-equiv="x-ua-compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link href="lib/bootstrap/css/bootstrap.min.css" rel="stylesheet">
<style>
        /*轮播广告*/
        .pic{
            margin: 0 auto;
            width: 800px;
            padding: 20px;
        }
        .carousel {
            background: white;
        }
        /*媒体查询：当视口小于 992px 时缩小了轮播 div 的宽度，图片换行*/
        @media (max-width:992px) {
            .pic{
                width: 415px;
            }
        }
</style>
</head>
<body>
<!--轮播广告-->
<div class="container hidden-xs " >
    <div id="myCarousel" class="carousel slide" data-ride="carousel">
        <!-- 轮播（Carousel）项目 -->
        <div class="carousel-inner ">
            <div class="item active">
                <div class="pic">
```

```
35                <img src="images/1p.jpg"  alt="">
36                <img src="images/2p.jpg"  alt="">
37                <img src="images/3p.png"  alt="">
38                <img src="images/4p.jpg"  alt="">
39                <img src="images/5p.jpg"  alt="">
40                <img src="images/6p.jpg"  alt="">
41             </div>
42          </div>
43          <div class="item ">
44             <div class="pic">
45                <img src="images/2p.jpg"  alt="">
46                <img src="images/3p.png"  alt="">
47                <img src="images/4p.jpg"  alt="">
48                <img src="images/5p.jpg"  alt="">
49                <img src="images/6p.jpg"  alt="">
50                <img src="images/7p.jpg"  alt="">
51             </div>
52          </div>
53          <div class="item">
54             <div class="pic">
55                <img src="images/3p.png"  alt="">
56                <img src="images/4p.jpg"  alt="">
57                <img src="images/5p.jpg"  alt="">
58                <img src="images/6p.jpg"  alt="">
59                <img src="images/7p.jpg"  alt="">
60                <img src="images/8p.jpg"  alt="">
61             </div>
62          </div>
63          <div class="item">
64             <div class="pic">
65                <img src="images/4p.jpg"  alt="">
66                <img src="images/5p.jpg"  alt="">
67                <img src="images/6p.jpg"  alt="">
68                <img src="images/7p.jpg"  alt="">
69                <img src="images/8p.jpg"  alt="">
70                <img src="images/1p.jpg"  alt="">
71             </div>
72          </div>
73          <div class="item ">
74             <div class="pic">
75                <img src="images/5p.jpg"  alt="">
76                <img src="images/6p.jpg"  alt="">
77                <img src="images/7p.jpg"  alt="">
78                <img src="images/8p.jpg"  alt="">
79                <img src="images/1p.jpg"  alt="">
80                <img src="images/2p.jpg"  alt="">
81             </div>
82          </div>
83          <div class="item ">
84             <div class="pic">
85                <img src="images/6p.jpg"  alt="">
86                <img src="images/7p.jpg"  alt="">
```

```
87                  <img src="images/8p.jpg"  alt="">
88                  <img src="images/1p.jpg"  alt="">
89                  <img src="images/2p.jpg"  alt="">
90                  <img src="images/3p.png"  alt="">
91              </div>
92          </div>
93          <div class="item">
94              <div class="pic">
95                  <img src="images/7p.jpg"  alt="">
96                  <img src="images/8p.jpg"  alt="">
97                  <img src="images/1p.jpg"  alt="">
98                  <img src="images/2p.jpg"  alt="">
99                  <img src="images/3p.png"  alt="">
100                 <img src="images/4p.jpg"  alt="">
101             </div>
102         </div>
103         <div class="item ">
104             <div class="pic">
105                 <img src="images/8p.jpg"  alt="">
106                 <img src="images/1p.jpg"  alt="">
107                 <img src="images/2p.jpg"  alt="">
108                 <img src="images/3p.png"  alt="">
109                 <img src="images/4p.jpg"  alt="">
110                 <img src="images/5p.jpg"  alt="">
111             </div>
112         </div>
113     </div>
114     <!-- 轮播（Carousel）导航 -->
115     <a class="carousel-control left  " href="#myCarousel" role="button "
          data-slide="prev">&lsaquo;</a>
116     <a class="carousel-control right " href="#myCarousel" role="button"
          data-slide="next">&rsaquo;</a>
117   </div>
118</div>
119<!--轮播广告结束-->
120<!-- jQuery (Bootstrap 的 JavaScript 插件需要引入 jQuery) -->
121<script src="lib/jquery/jquery-1.11.0.min.js"></script>
122<!-- 包括所有已编译的插件 -->
123<script src="lib/bootstrap/js/bootstrap.min.js"></script>
124</body>
125</html>
```

【任务 6-整合所有模块，完成 footer 部分】

【任务描述】

餐饮网站重要的模块部分已经全部完成了，接下来我们要完成所有模块的整合，并且为网页添加 footer 部分，该网页 footer 部分的页面效果如图 8-52 所示。

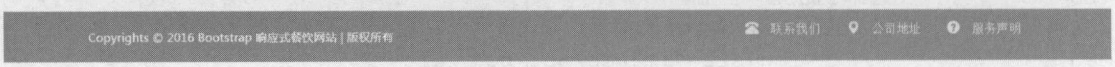


图8-52　footer部分

【任务分析】

在前面的任务中完成的每个模块都可以单独查看效果，那么整合的工作只需要把这些模块按照在网页中的结构粘贴代码即可。该网页的整体页面结构如图 8-53 所示。

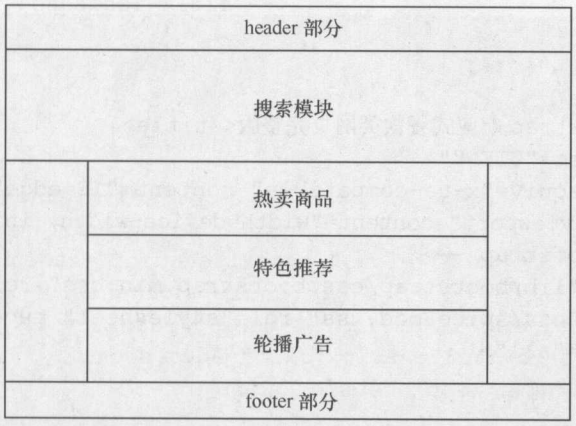


图8-53　餐饮网页整体页面结构图

图 8-53 所示的餐饮网页的模块分布从上至下的顺序是 header 部分、搜索模块、热卖商品、特色推荐、轮播广告、footer 部分。

footer 部分的页面结构如图 8-54 所示。

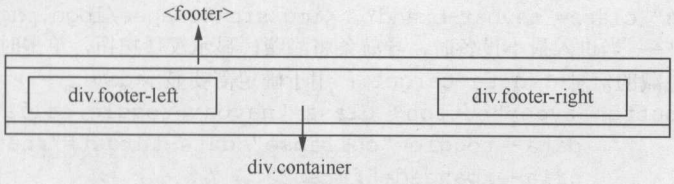


图8-54　餐饮footer部分结构图

图 8-54 所示的 footer 部分最外层使用<footer>标签，在该标签内嵌套 标签并使用.container 类，然后分别添加左侧和右侧的 标签进行布局。该页面的实现细节具体分析如下。

（1）在左侧 标签中嵌套<p>标签添加描述信息。

（2）在右侧 标签中使用<ul>列表嵌套<a>标签，添加一些公司地址和联系方式的链接，这里也使用了 Bootstrap 的字体图标。3 个图标对应的类如下。

- glyphicon glyphicon-phone-alt。
- glyphicon glyphicon-map-marker。
- glyphicon glyphicon-question-sign。

（3）整合整个页面，在整合的过程中，需要将 HTML 代码和 CSS 代码分成两个文件，再将其分别按照顺序粘贴到对应文件即可。如果出现重复的 CSS 代码，可以采用合写的方式。

【代码实现】

对页面的结构有所了解后，即可开始编写代码来实现它。该项目完整的 HTML 页面代码如 code\08\0801\0801.html 所示。

0801.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Bootstrap 响应式餐饮类网页完整版</title>
    <meta charset="UTF-8">
    <meta http-equiv="x-ua-compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- 引入 Bootstrap -->
    <link href="lib/bootstrap/css/bootstrap.min.css" rel="stylesheet">
    <link href="css/spiceFood.css" rel="stylesheet" type="text/css"
          media="all">
</head>
<body>
<!-- header -->
<header>
<div class="head-top"></div>
<nav class="navbar navbar-default" >
    <div class="container-fluid">
        <div class="container">
            <div class="navbar-header">
  <a href="#" class="navbar-brand" ><img src="images/logo.png"  alt=""></a>
            <!-- 当进入最小设备时，导航条将隐藏，显示汉堡按钮，单击时可以切面显示导航条
            隐藏的信息 * data-target : 用于确定需要显示 div   -->
            <button type="button" class="navbar-toggle collapsed"
                    data- toggle="collapse" data-target="#navbar-collapse"
                    aria-expanded="false">
                <span class="sr-only">汉堡按钮</span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
        </div>
        <!-- 导航链接、表单和其他内容切换 -->
        <div class="collapse navbar-collapse" id="navbar-collapse">
              <ul class="nav navbar-nav">
                  <li><a href="#" >主页</a></li>
                  <li><a href="#">餐厅</a></li>
                  <li><a href="#">健康</a></li>
                  <li><a href="#">话题</a></li>
                  <li><a href="#">联系我们</a></li>
              </ul>
               <div class="navbar-right ">
                  <a href="#">
```

```
40      <h3> <span >￥0.00 <img src="images/bag.png" alt=""/></span></h3>
41                  </a>
42              </div>
43          </div>
44      </div>
45    </div><!-- /.navbar-collapse -->
46    </div><!-- /.container-fluid -->
47 </nav>
48 </header>
49 <!-- header -->
50 <!--搜索区域-->
51 <div class="search" >
52    <div class="container">
53       <div class="reservation">
54          <form class="form-horizontal" role="form">
55             <div class="form-group">
56                <div class="col-sm-12 col-md-12 col-lg-12">
57                   <input type="text" class="form-control input-lg"
                            id="name" placeholder="请输入餐厅名称">
58                </div>
59             </div>
60             <div class="form-group">
61                <div class="col-sm-12 col-md-12 col-lg-12">
62                   <select id="country"  class="form-control input-lg">
63                      <option value="null">请选择城市</option>
64                      <option value="bj">北京</option>
65                      <option value="sh">上海</option>
66                      <option value="sz">深圳</option>
67                   </select>
68                </div>
69             </div>
70             <div class="form-group">
71                <div class="searchbtn">
72                   <button type="submit" class="btn btn-success btn-lg">
73                  <img src="images/search-icon.png"> 搜索</button>
74                </div>
75             </div>
76          </form>
77       </div>
78    </div>
79 </div>
80 <!-- 搜索区域结束 -->
81 <!-- 热卖商品 -->
82 <div class="hot">
83    <div class="container">
84       <div class="col-md-4 ">
85          <h3>米西奈斯煎饼</h3>
86          <img src="images/4.jpg" class="img-responsive" alt="">
87          <div>
88             <p class="glyphicon glyphicon-thumbs-up">两种口味可供选择 </p>
```

```
89                <div class="cur">
90                    <span ><a class="morebtn" href="#">添加到购物车</a></span>
91                      <span><h6>一口价 ： ￥45.00</h6></span>
92                </div>
93             </div>
94          </div>
95          <div class="col-md-4">
96             <h3>蒙特斯大虾</h3>
97             <img src="images/1.jpg" class="img-responsive" alt="蒙特斯大虾">
98             <div>
99                <p class="glyphicon glyphicon-thumbs-up">两种口味可供选择 </p>
100               <div class="cur">
101                   <span><a class="morebtn" href="#">添加到购物车</a></span>
102                       <span><h6>一口价 ： ￥55.00</h6></span>
103                </div>
104             </div>
105         </div>
106         <div class="col-md-4">
107            <h3>香酥鸡排</h3>
108            <img src="images/3.jpg" class="img-responsive" alt="香酥鸡排">
109            <div>
110               <p class="glyphicon glyphicon-thumbs-up">两种口味可供选择 </p>
111               <div class="cur">
112                   <span><a class="morebtn " href="#">添加到购物车</a></span>
113                     <span><h6>一口价 ： ￥65.00</h6></span>
114               </div>
115            </div>
116         </div>
117      </div>
118</div>
119<!--热卖商品结束 -->
120<!--特色推荐-->
121<div class="container">
122   <div class="choose">
123      <div class="row">
124         <div class="col-md-12">
125            <div class="navbar-header hidden-xs">
126       <a class="navbar-brand" href="#"><img src="images/title.jpg" /></a>
127            </div>
128            <!-- Nav tabs -->
129            <ul class="nav nav-pills navbar-right " role="tablist"
                  style="margin-right: 0px;">
130               <li role="presentation" class="active"><a href="#dishes"
                     role="tab" data-toggle="tab">菜品</a></li>
131               <li role="presentation"><a href="#drink"  role="tab"
                     data-toggle="tab">饮品</a></li>
132               <li role="presentation"><a href="#staple"  role="tab"
                     data-toggle="tab">主食</a></li>
133            </ul>
134         </div>
```

```
    </div>
    <div class="row">
        <div class="col-md-12">
            <div class="tab-content">
                <!--菜品-->
                <div role="tabpanel" class="tab-pane active" id="dishes">
                    <div class=" col-md-2 col-sm-4 col-xs-6">
                        <img src="images/products/small01.jpg"/>
                        <p>菜品</p>
                        <p>￥145.0</p>
                    </div>
                    <div class="product-item col-md-2 col-sm-4 col-xs-6">
                        <img src="images/products/small02.jpg"/>
                        <p>菜品</p>
                        <p>￥165.0</p>
                    </div>
                    <div class="product-item col-md-2 col-sm-4 col-xs-6">
                        <img src="images/products/small04.jpg"/>
                        <p>菜品</p>
                        <p>￥165.0</p>
                    </div>
                    <div class="product-item col-md-2 col-sm-4 col-xs-6">
                        <img src="images/products/small02.jpg"/>
                        <p>菜品</p>
                        <p>￥165.0</p>
                    </div>
                    <div class="product-item col-md-2 col-sm-4 col-xs-6">
                        <img src="images/products/small07.jpg"/>
                        <p>菜品</p>
                        <p>￥145.0</p>
                    </div>
                    <div class="product-item col-md-2 col-sm-4 col-xs-6">
                        <img src="images/products/small08.jpg"/>
                        <p>菜品</p>
                        <p>￥165.0</p>
                    </div>
                </div>

                <!--饮品-->
                <div role="tabpanel" class="tab-pane" id="drink">
                    <div class="product-item col-md-2 col-sm-4 col-xs-6">
                        <img src="images/products/small03.jpg"/>
                        <p>饮品</p>
                        <p>￥98.0</p>
                    </div>
                    <div class="product-item col-md-2 col-sm-4 col-xs-6">
                        <img src="images/products/small03.jpg"/>
                        <p>饮品</p>
                        <p>￥98.0</p>
                    </div>
```

```
                </div>
                <!--</div>-->
                <!--主食-->
                <div role="tabpanel" class="tab-pane" id="staple">
                    <div class="product-item col-md-2 col-sm-4 col-xs-6">
                        <img src="images/products/small05.jpg"/>
                        <p>主食</p>
                        <p>¥56.0</p>
                    </div>
                    <div class="product-item col-md-2 col-sm-4 col-xs-6">
                        <img src="images/products/small06.jpg"/>
                        <p>主食</p>
                        <p>¥56.0</p>
                    </div>
                    <div class="product-item col-md-2 col-sm-4 col-xs-6">
                        <img src="images/products/small09.jpg"/>
                        <p>主食</p>
                        <p>¥56.0</p>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
<!--特色推荐结束-->
<!--轮播广告-->
<div class="container hidden-xs" >
    <div id="myCarousel" class="carousel slide" data-ride="carousel">
        <!-- 轮播（Carousel）项目 -->
        <div class="carousel-inner ">
            <div class="item active">
                <div class="pic">
                    <img src="images/1p.jpg"  alt="">
                    <img src="images/2p.jpg"  alt="">
                    <img src="images/3p.png"  alt="">
                    <img src="images/4p.jpg"  alt="">
                    <img src="images/5p.jpg"  alt="">
                    <img src="images/6p.jpg"  alt="">
                </div>
            </div>
            <div class="item ">
                <div class="pic">
                    <img src="images/2p.jpg"  alt="">
                    <img src="images/3p.png"  alt="">
                    <img src="images/4p.jpg"  alt="">
                    <img src="images/5p.jpg"  alt="">
                    <img src="images/6p.jpg"  alt="">
                    <img src="images/7p.jpg"  alt="">
                </div>
```

```
235         </div>
236         <div class="item">
237            <div class="pic">
238               <img src="images/3p.png"  alt="">
239               <img src="images/4p.jpg"  alt="">
240               <img src="images/5p.jpg"  alt="">
241               <img src="images/6p.jpg"  alt="">
242               <img src="images/7p.jpg"  alt="">
243               <img src="images/8p.jpg"  alt="">
244            </div>
245         </div>
246         <div class="item">
247            <div class="pic">
248               <img src="images/4p.jpg"  alt="">
249               <img src="images/5p.jpg"  alt="">
250               <img src="images/6p.jpg"  alt="">
251               <img src="images/7p.jpg"  alt="">
252               <img src="images/8p.jpg"  alt="">
253               <img src="images/1p.jpg"  alt="">
254            </div>
255         </div>
256         <div class="item ">
257            <div class="pic">
258               <img src="images/5p.jpg"  alt="">
259               <img src="images/6p.jpg"  alt="">
260               <img src="images/7p.jpg"  alt="">
261               <img src="images/8p.jpg"  alt="">
262               <img src="images/1p.jpg"  alt="">
263               <img src="images/2p.jpg"  alt="">
264            </div>
265         </div>
266         <div class="item ">
267            <div class="pic">
268               <img src="images/6p.jpg"  alt="">
269               <img src="images/7p.jpg"  alt="">
270               <img src="images/8p.jpg"  alt="">
271               <img src="images/1p.jpg"  alt="">
272               <img src="images/2p.jpg"  alt="">
273               <img src="images/3p.png"  alt="">
274            </div>
275         </div>
276         <div class="item">
277            <div class="pic">
278               <img src="images/7p.jpg"  alt="">
279               <img src="images/8p.jpg"  alt="">
280               <img src="images/1p.jpg"  alt="">
281               <img src="images/2p.jpg"  alt="">
282               <img src="images/3p.png"  alt="">
283               <img src="images/4p.jpg"  alt="">
284            </div>
```

```
285          </div>
286          <div class="item ">
287             <div class="pic">
288                <img src="images/8p.jpg"  alt="">
289                <img src="images/1p.jpg"  alt="">
290                <img src="images/2p.jpg"  alt="">
291                <img src="images/3p.png"  alt="">
292                <img src="images/4p.jpg"  alt="">
293                <img src="images/5p.jpg"  alt="">
294             </div>
295          </div>
296       </div>
297       <!-- 轮播（Carousel）导航 -->
298       <a class="carousel-control left  " href="#myCarousel" role="button "
             data-slide="prev">&lsaquo;</a>
299       <a class="carousel-control right " href="#myCarousel" role="button"
             data-slide="next">&rsaquo;</a>
300    </div>
301</div>
302<!--轮播广告结束-->
303<!-- footer-->
304<footer class="footer">
305    <div class="container">
306       <div class="footer-left">
307  <p>Copyrights © 2016 Bootstrap 响应式餐饮网站  | 版权所有 <a href="#"> </a></p>
308       </div>
309       <div class="footer-right">
310          <ul>
311             <li><a href="#"><i class="glyphicon
                    glyphicon-phone-alt">  联系我们</i></a></li>
312             <li><a href="#"><i class="glyphicon
                    glyphicon-map-marker">  公司地址</i></a></li>
313             <li><a href="#"><i class="glyphicon
                    glyphicon-question- sign"> 服务声明</i></a></li>
314
315          </ul>
316       </div>
317    </div>
318</footer>
319<!-- footer-->
320<!-- jQuery (Bootstrap 的 JavaScript 插件需要引入 jQuery) -->
321<script src="lib/jquery/jquery-1.11.0.min.js"></script>
322<!-- 包括所有已编译的插件 -->
323<script src="lib/bootstrap/js/bootstrap.min.js"></script>
324</body>
325</html>
```

该项目完整的 CSS 代码如 code\08\0801\css\spiceFood.css 所示。

spiceFood.css

```
1  /*第 8 单元 项目 8-1 Bootstrap 餐饮类网站首页*/
```

```
2
3  /*为页面中以下元素设置字体为 Microsoft YaHei*/
4  body,button, input, select, h3,  h6 { font-family: Microsoft YaHei}
5  /*header 部分*/
6  .head-top{
7     background: #5fa022;
8     padding: 0.8em 0;
9  }
10 .navbar-brand{
11    padding: 0 0;
12 }
13 /*设置图片 logo 的大小和位置*/
14 .navbar-brand>img{
15    height: auto;
16    margin-right: 5px;
17    margin-top: 5px;
18    width: 250px;
19 }
20 /*设置整个导航菜单的内边距、背景色和阴影*/
21 .navbar-default{
22    padding: 1.5em 0;
23    background-color: #f2f0f1;
24    box-shadow: 12px -5px 39px -12px;
25 }
26 /*设置导航栏中菜单 a 链接的样式*/
27 .navbar-default .navbar-nav>li a {
28    top: 10px;
29    padding: 0.5em 3em;
30    text-decoration: none;
31    font-weight: 600;
32    font-size: 1.2em;
33    color:#919191;
34 }
35 /*设置导航栏菜单的鼠标悬停和获取焦点时的状态*/
36 .navbar-default .navbar-nav> li >a:hover,
37 .navbar-default .navbar-nav> li >a:focus {
38    background:#D96B66;
39    color: white;
40    border-radius: 3px;
41    -webkit-border-radius: 3px;
42 }
43 /*设置导航栏右侧 a 链接（0.00）的颜色*/
44 .navbar-right>a{
45    color:#D96B66;
46 }
47 /*媒体查询：当视口小于 970px 时，导航菜单字体变小，避免了换行的问题*/
48 @media (max-width: 970px) {
49    .navbar-default .navbar-nav>li a {
50       font-size: 1.1em;
51    }
```

```
52 }
53 /*header 部分结束*/
54 /*搜索模块*/
55 /*设置搜索按钮前面的搜索图标*/
56 .search{
57     background-image: url(../images/banner.jpg);
58     background-size: cover;
59     -webkit-background-size: cover;
60     min-height: 600px;
61     margin-top: -20px;
62 }
63 /*设置搜索框外层的 div 样式，形成白色透明背景*/
64 .reservation {
65     padding: 60px 60px;
66     width:50%;
67     background: rgba(255, 255, 255, 0.7);
68     margin: 0 auto;
69     margin-top:15%;
70     font-weight: 500;
71     color: #f2f0f1;
72     font-size: 1.2em;
73     outline: none;
74 }
75 /*设置搜索按钮的样式*/
76 .btn{
77     width: 50% ;
78     background: #D96B66;
79     color: #fff;
80     padding: 5px;
81     border: none;
82     border-radius: 4px;
83     -webkit-border-radius: 4px;
84 }
85 .form-control{
86     color: #8e908d;
87 }
88 .searchbtn{
89     text-align: center;
90 }
91 /*媒体查询：当屏幕小于 768px 时搜索表单白色背景大小发生变化*/
92 @media (max-width: 768px){
93     .reservation {
94         padding: 20px 20px;
95         width: 90%;
96     }
97 }
98 /*搜索模块结束*/
99 /*热卖商品*/
100.hot h3 {
101    margin-top: -20px;
```

```
102    color: #1A1A1A;
103    font-size: 1.5em;
104    font-weight: 600;
105    margin: 0 0 1em;
106    padding: 0 0 0.5em;
107    border-bottom: 2px solid #eee;
108}
109.hot p {
110    color: #5fa022;;
111    font-size: 1em;
112    font-weight: 400;
113    line-height: 1.8em;
114    margin: 1em 0;
115}
116.hot {
117    padding: 3em 0;
118    border: 1px solid #eee;
119    margin: 0 0;
120}
121.hot h6 {
122    color: #C15162;
123    font-size: 1.5em;
124    font-weight: 400;
125    margin: 0.3em 0;
126}
127a.morebtn {
128    display: block;
129    font-size: 1em;
130    color: #FFF;
131    text-decoration: none;/*去掉 a 链接的下划线*/
132    font-weight: 400;
133    background: #D96B66;
134    padding: 10px 18px;
135    transition: 0.5s all ease;
136    -webkit-transition: 0.5s all ease;
137    border-radius: 3px;
138    -webkit-border-radius: 3px;
139}
140a.morebtn:hover{
141    background:  #5fa022;
142}
143@media (max-width: 1024px) {
144    a.morebtn{
145        max-width:410px;/*当视口小于 1024px 时按钮的最大宽度*/
146    }
147}
148/*热卖商品结束*/
149/*特色推荐*/
150/*设置左上角图片的位置和宽高*/
151.navbar-brand>img{
```

```
    height: auto;
    width:250px;
    margin-right: 5px;
    margin-top: 5px;
}
/*设置选项卡菜单的字体颜色*/
.nav-pills>li>a{
    color:#8e908d ;
}
/*设置选项卡菜单鼠标悬停和获取焦点时的背景色和字体颜色*/
.nav-pills>li.active>a, .nav-pills>li.active>a:focus,
  .nav-pills>li.active>a:hover {
    color: #fff;
    background-color: #5A9522;
}
.choose{
    border: 1px solid silver;
}
/*设置选项卡内容的位置*/
.tab-content{
    margin: 5px;
    text-align: center;/*居中显示*/
}
/*特色推荐结束*/
/*轮播广告*/
.pic{
    margin: 0 auto;
    width: 800px;
    padding: 20px;
}
.carousel {
    background: white;
}
/*媒体查询：当视口小于 992px 时缩小了轮播 div 的宽度，图片换行*/
@media (max-width:992px) {
    .pic{
        width: 415px;
    }
}
/*轮播广告结束*/
/*footer 部分*/
.footer p {
    color: #fff;
    font-size: 1em;
    line-height: 1.8em;
    vertical-align: middle;
    margin: 0.4em 0;
}
.footer p a{
    color: #fff;
    text-decoration:none;
}
```

```
.footer {
    padding: 1em 0;
    background-color:#D96B66;
}
.footer-right ul {
    padding: 0;
}
.footer-right li {
    display: inline-block;
    margin: 0 1em;
}
.footer-left {
    float: left;
}
.footer-right {
    float: right;
}
/*设置字体图标的大小和颜色*/
.glyphicon{
    font-size: 1.2em;
    color: #f2f0f1 ;
}
```

【项目总结】

本项目的练习重点：

本项目主要练习的知识点是 Bootstrap 的布局容器、导航栏、表单、按钮、栅格系统、标签页、响应式工具以及轮播插件的使用。

本项目的练习方法：

建议读者在编码时，按照顺序分模块完成，最后参考完整代码将各模块进行整合。

在学习本单元内容时，建议读者先熟悉 Bootstrap 手册，熟悉 Bootstrap 提供的样式。熟悉 Bootstrap 后，读者可以尝试在本项目中增加其他模块。

本项目的注意事项：

本项目的每个任务模块代码都可独立运行，与其他模块没有耦合。如果在整合时遇到问题，可以检查每个独立模块的代码是否正确，然后对错误的代码进行有针对性的修改。

【思考题】

1. 请简述 Bootstrap 包中提供了哪些内容以及它们的作用。
2. 请简述 Bootstrap 有哪些特点。

Responsive Web Design

Appendix 1

附录 1
HTML5 新增标签和废除标签

1. HTML5 新增标签

标签名称	标签描述
<article>	定义独立的内容
<aside>	定义两栏或多栏页面的侧边栏内容
<audio>	定义音频内容
<bdi>	定义文本的文本方向，使其脱离其周围文本的方向设置
<canvas>	定义图形，如图表和其他图像。<canvas>元素只是图形容器（画布），必须使用脚本来绘制图形
<command>	表示命令按钮，如单选按钮、复选框或按钮。只有当 command 元素位于 menu 元素内时，该元素才是可见的；否则不会显示这个元素，但是可以用它规定键盘快捷键
<datalist>	定义选项列表，需与 input 元素配合使用，通过 input 元素的 list 属性来绑定，用来定义 input 可能的值
<details>	用于描述文档或文档某个部分的细节
<dialog>	定义对话框或窗口
<embed>	定义外部交互内容或插件
<figcaption>	定义 figure 元素的标题
<figure>	定义一组媒体内容（图像、图表、照片、代码等）以及它们的标题
<footer>	定义一个页面或一个区域的页脚
<header>	定义一个页面或一个区域的头部
<keygen>	定义表单里一个生成的键值。规定用于表单的密钥对生成器字段
<mark>	定义有标记的文本
<meter>	定义预定义范围内的度量
<nav>	定义导航链接
<output>	定义不同类型的输出
<progress>	定义任何类型任务的进度
<rp>	定义若浏览器不支持 ruby 元素时显示的内容
<rt>	定义 ruby 注释的解释
<ruby>	定义 ruby 注释（中文注音或字符）
<section>	定义页面中的一个内容区块，如章节、页眉、页脚或页面的其他部分
<source>	为媒介元素（如<video>和<audio>）定义媒介资源
<summary>	为<details>元素定义可见的标题
<time>	定义日期/时间
<track>	定义用在媒体播放器中的文本轨道
<video>	定义视频
<wbr>	表示软换行，即当浏览器窗口或父级元素足够宽时不换行，而宽度不够时自动在此处换行

2. HTML5 废除标签

标签名称	标签描述
<basefont>	定义基准字体
<big>	呈现大号字体效果
	对其包围的文本进行水平居中处理
<font>	规定文本字体、大小和颜色
<strike>	可定义加删除线文本定义
<tt>	呈现类似打字机或者等宽的文本效果
<u>	为文本添加下划线
<frame>	定义<frameset>中的一个特定的窗口（框架）
<frameset>	定义一个框架集
<noframes>	为那些不支持框架的浏览器显示文本
<applet>	定义内嵌对象，使用<object>标签代替
<bgsound>	在文档中插入背景音乐
<blink>	定义在浏览器中显示闪烁的文本
<marquee>	定义多种滚动效果
<rb>	设定被标示的元素对象，使用<ruby>代替
<acronym>	定义只取首字母缩写，使用<abbr>代替
<dir>	定义目录列表，使用 代替
<isindex>	定义单行的文字提交输入框，可以用<form>与<input>相结合的方式代替
<listing>	定义预格式文本，使用<pre>代替
<xmp>	预定义格式文字显示，使用<code>替代
<nextid>	创建编辑软件可以读取的唯一标识符，使用<guids>代替
<plaintex>	定义简单文字，使用</plian>代替

Responsive Web Design

Appendix 2

附录 2
CSS3 新增属性

1. 动画属性（Animation）

属性	描述	值	属性值说明
@keyframes	创建动画	animationname	必需，定义动画的名称
		keyframes-selector	必需，动画时长的百分比 合法的值： • 0%～100% • from（与 0%相同） • to（与 100%相同）
		css-styles	必需，一个或多个合法的 CSS 样式属性
animation	是一个简写属性，用于设置 6 个动画属性	animation-name	规定@keyframes 动画的名称
		animation-duration	规定完成动画所花费的时间，以秒或毫秒计
		animation-timing-function	规定动画的速度曲线
		animation-delay	规定在动画开始之前的延迟
		animation-iteration-count	规定动画应该播放的次数
		animation-direction	规定是否应该轮流反向播放动画
animation -name	定义@keyframes 动画的名称	keyframename	规定需要绑定到选择器的 keyframe 的名称
		none	规定无动画效果（可用于覆盖来自级联的动画）
animation-duration	定义动画完成一个周期所需的时间	time	以秒或毫秒计，默认值是 0，意味着没有动画效果
animation-timing-function	定义动画的速度曲线	linear	动画从头到尾的速度是相同的
		ease	默认，动画以低速开始，然后加快，在结束前变慢
		ease-in	动画以低速开始
		ease-out	动画以低速结束
		ease-in-out	动画以低速开始和结束
animation-delay	定义动画开始前等待的时间	time	可选，以秒或毫秒计，默认值是 0
animation-iteration-count	定义动画的播放次数	n	定义动画播放次数的数值
		infinite	规定动画应该无限次播放
animation-direction	规定动画是否在下一周期逆向地播放	normal	默认值，动画应该正常播放
		alternate	动画应该轮流反向播放
animation-play-state	规定动画正在运行还是暂停	paused	规定动画已暂停
		running	规定动画正在播放
animation-fill-mode	规定动画在播放之前或之后，其动画效果是否可见	none	不改变默认行为
		forwards	当动画完成后，保持最后一个属性值（在最后一个关键帧中定义）

续表

属性	描述	值	属性值说明
animation-fill-mode	规定动画在播放之前或之后，其动画效果是否可见	backwards	在 animation-delay 所指定的一段时间内，在动画显示之前，应用开始属性值（在第一个关键帧中定义）
		both	向前和向后填充模式都被应用

2. 背景属性（Background）

属性	描述	值	属性值说明
background-clip	规定背景的绘制区域	border-box	背景被裁剪到边框盒
		padding-box	背景被裁剪到内边距框
		content-box	背景被裁剪到内容框
background-origin	规定背景图片的定位区域	padding-box	背景图像相对于内边距框来定位
		border-box	背景图像相对于边框盒来定位
		content-box	背景图像相对于内容框来定位
background-size	规定背景图片的尺寸	length	设置背景图像的高度和宽度； 第一个值设置宽度，第二个值设置高度。如果只设置一个值，则第二个值会被设置为“auto”
		percentage	以父元素的百分比来设置背景图像的宽度和高度； 第一个值设置宽度，第二个值设置高度。如果只设置一个值，则第二个值会被设置为“auto”
		cover	把背景图像扩展至足够大，以使背景图像完全覆盖背景区域； 背景图像的某些部分也许无法显示在背景定位区域中
		contain	把图像扩展至最大尺寸，以使其宽度和高度完全适应内容区域

3. 边框属性（Border 和 Outline）

属性	描述	值	属性值说明
border-bottom-left-radius	定义边框左下角的圆角半径	length	具体的像素数值
		%	百分比值
border-bottom-right-radius	定义边框右下角的圆角半径	length	具体的像素数值
		%	百分比值
border-image	简写属性，设置所有 border-image-*属性	border-image-source	用在边框的图片的路径
		border-image-slice	图片边框向内偏移
		border-image-width	图片边框的宽度
		border-image-outset	边框图像区域超出边框的量
		border-image-repeat	图像边框是否应平铺、铺满或拉伸
border-image-outset	规定边框图像区域超出边框的量	length	具体的像素数值
		number	代表对应的 border-width 的倍数

续表

属性	描述	值	属性值说明
border-image-repeat	图像边框是否应平铺、铺满或拉伸	repeat	平铺（重复）图像来填充区域
		round	铺满，类似 repeat 值，如果无法完整平铺所有图像，则对图像进行缩放以适应区域
		stretch	拉伸图像来填充区域
border-image-slice	规定图像边框的向内偏移	number	数字值，代表图像中的像素（如果是光栅图像）或矢量坐标（如果是矢量图像）
		%	相对于图像尺寸的百分比值：图像的宽度影响水平偏移，高度影响垂直偏移
		fill	保留边框图像的中间部分
border-image-source	规定用来作为边框的图片	none	不使用图像
		image	用来作为边框的图像的路径
animation-direction	规定动画是否在下一周期逆向地播放	normal	默认值，动画应该正常播放
		alternate	动画应该轮流反向播放
border-image-width	规定图片边框的宽度	number	代表对应的 border-width 的倍数
		%	参考边框图像区域的尺寸：区域的高度影响水平偏移，宽度影响垂直偏移
		auto	如果规定该属性，则宽度为对应的图像切片的固有宽度
border-radius	简写属性，用于设置 4 个 border-*-radius 属性	length	具体的像素数值
		%	百分比值
border-top-left-radius	定义边框左上角的圆角半径	length	具体的像素数值
		%	百分比值
border-top-right-radius	定义边框右上角的圆角半径	length	具体的像素数值
		%	百分比值
box-shadow	向元素添加一个或多个阴影	h-shadow	必需，水平阴影的位置，允许负值
		v-shadow	必需，垂直阴影的位置，允许负值
		blur	可选，模糊距离
		spread	可选，阴影的尺寸
		color	可选，阴影的颜色，请参阅 CSS 颜色值
		inset	可选，将外部阴影（outset）改为内部阴影

4. Box 属性

属性	描述	值	属性值说明
overflow-x	如果内容溢出了元素内容区域，是否对内容的左/右边缘进行裁剪	visible	不裁剪内容，可能会显示在内容框之外
		hidden	裁剪内容，不提供滚动机制
		scroll	裁剪内容，提供滚动机制

续表

属性	描述	值	属性值说明
overflow-x	如果内容溢出了元素内容区域，是否对内容的左/右边缘进行裁剪	auto	如果溢出框，则应该提供滚动机制
		no-display	如果内容不适合内容框，则删除整个框
		no-content	如果内容不适合内容框，则隐藏整个内容
overflow-y	如果内容溢出了元素内容区域，是否对内容的上/下边缘进行裁剪	visible	不裁剪内容，可能会显示在内容框之外
		hidden	裁剪内容，不提供滚动机制
		scroll	裁剪内容，提供滚动机制
		auto	如果溢出框，则应该提供滚动机制
		no-display	如果内容不适合内容框，则删除整个框
		no-content	如果内容不适合内容框，则隐藏整个内容
overflow-style	规定溢出元素的首选滚动方法	auto	默认值
		scrollbar	为溢出元素添加滚动条
		panner	显示内部元素的一部分，并可以移动该内部元素
		move	用户能够直接移动元素的内容。通常，用户能够用鼠标拖动内容
		marquee	内容自主移动，不需任何用户代理对其控制
rotation	围绕由 rotation-point 属性定义的点对元素进行旋转	angle	元素旋转角度，可能的值为 0～360deg
rotation-point	定义距离上左边框边缘的偏移点	left top left center left bottom right top right center right bottom center top center center center bottom	如果只规定一个关键词，则第二个值将是“center”
		x% y%	百分比值，参考边框盒宽度和高度

5. Color 属性

属性	描述	值	属性值说明
gradient	渐变效果	linear-gradient	线性渐变
		radial Gradient	径向渐变
opacity	设置元素的不透明级别	value	规定不透明度，从 0.0（完全透明）到 1.0（完全不透明）
		inherit	从父元素继承 opacity 属性的值
rendering-intent	允许使用颜色配置文件渲染意图的默认以外的规范		
color-profile	允许使用源的颜色配置文件的默认以外的规范		

6. Content for Paged Media 属性

属性	描述
bookmark-label	规定书签的标记
bookmark-level	规定书签的级别
bookmark-target	规定书签链接的目标
float-offset	将元素放在 float 属性通常放置的位置的相反方向
hyphenate-after	规定连字单词中连字符之后的最小字符数
hyphenate-before	规定连字单词中连字符之前的最小字符数
hyphenate-character	规定当发生断字时显示的字符串
hyphenate-lines	指示元素中连续断字连线的最大数
hyphenate-resource	规定帮助浏览器确定断字点的外部资源（逗号分隔的列表）
hyphens	设置如何对单词进行拆分，以改善段落的布局
image-resolution	规定图像的正确分辨率
marks	向文档中添加裁切标记或十字标记
string-set	接受一个逗号分隔的命名字符串列表，每个命名的字符串后面跟着一个内容列表，该列表指定要哪些文本复制到已命名的字符串中

7. 可伸缩框属性（Flexible Box）

属性	描述	值	属性值说明
box-align	规定如何对齐框的子元素	start	对于正常方向的框，每个子元素的上边缘沿着框的顶边放置； 对于反方向的框，每个子元素的下边缘沿着框的底边放置
		end	对于正常方向的框，每个子元素的下边缘沿着框的底边放置； 对于反方向的框，每个子元素的上边缘沿着框的顶边放置
		center	均等地分割多余的空间，一半位于子元素之上，另一半位于子元素之下
		baseline	如果box-orient 是inline-axis或horizontal，所有子元素均与其基线对齐
		stretch	拉伸子元素以填充包含块
box-direction	规定框元素的子元素以什么方向来排列	normal	以默认方向显示子元素
		reverse	以反方向显示子元素
		inherit	应该从子元素继承 box-direction 属性的值
box-flex	规定框的子元素是否可伸缩	value	元素的可伸缩行。柔性是相对的，如 box-flex 为 2 的子元素两倍于 box-flex 为 1 的子元素
box-flex-group	将可伸缩元素分配到柔性分组	integer	一个整数（第一个柔性分组是 1，后面的柔性分组是更大的值）
box-lines	规定当超出父元素框的空间时，是否换行显示	single	所有子元素会被放置在单独的行或列中（无法匹配的元素会被视为溢出）
		multiple	允许框扩展为多行，以容纳其所有子元素

续表

属性	描述	值	属性值说明
box-ordinal-group	规定框的子元素的显示次序	integer	一个整数，指示子元素的显示次序
box-orient	规定框的子元素是否应水平或垂直排列	horizontal	在水平行中从左向右排列子元素
		vertical	从上向下垂直排列子元素
		inline-axis	沿着行内轴来排列子元素（映射为 horizontal）
		block-axis	沿着块轴来排列子元素（映射为 vertical）
		inherit	应该从父元素继承 box-orient 属性的值
box-pack	规定水平框中的水平位置或者垂直框中的垂直位置	start	对于正常方向的框，首个子元素的左边缘被放在左侧（最后的子元素后是所有剩余的空间）； 对于相反方向的框，最后的子元素的右边缘被放在右侧（首个子元素前是所有剩余的空间）
		end	对于正常方向的框，最后的子元素的右边缘被放在右侧（首个子元素前是所有剩余的空间）； 对于相反方向的框，首个子元素的左边缘被放在左侧（最后的子元素后是所有剩余的空间）
		center	均等地分割多余空间，其中一半空间被置于首个子元素前，另一半空间被置于最后一个子元素后
		justify	在每个子元素之间分割多余的空间（首个子元素前和最后一个子元素后没有多余的空间）

8. 内容生成（Generated Content）

属性	描述
crop	允许被替换元素仅仅是对象的矩形区域，而不是整个对象
move-to	从流中删除元素，然后在文档中后面的点上重新插入
page-policy	确定元素基于页面的 occurrence 应用于计数器还是字符串值

9. Grid 属性

属性	描述	值	属性值说明
grid-columns	规定网格中每个列的宽度	length	具体像素值
		%	百分比值
		none	无
		inherit	从父元素继承
grid-rows	规定网格中每个列的高度	length	具体像素值
		%	百分比值
		none	无
		inherit	从父元素继承

10. Hyperlink 属性

属性	描述	值	属性值说明
target	简写属性，设置 target-name、target-new 以及 target-position 属性	target-name	规定在何处打开超链接（target destination）
		target-new	规定应该在新窗口或已有窗口的新标签页中打开超链接
		target-position	规定在何处放置新的目的地链接
target-name	规定在何处打开链接（链接的目标）	current	在链接所在的框架、标签页或窗口中打开超链接
		root	在当前标签页或窗口中打开超链接
		parent	在父框架中打开超链接。如果当前框架没有父框架，则将该值视为 root
		new	创建新的目的地
		modal	在新的（暂时创建的）模态窗口中打开新窗口
		name	在已有框架、窗口或标签页中打开超链接。如果 name 目的地不存在，则用该名称创建新的目的地
target-new	规定目标链接在新窗口还是在已有窗口的新标签页中打开	window	在新窗口中打开超链接
		tab	在已有窗口的新标签页中打开超链接
		none	不创建新的目的地
target-position	规定在何处放置新的目标链接	above	在当前标签页/窗口之前放置新的目的地标签页/窗口
		behind	在当前标签页/窗口之后放置新的目的地标签页/窗口
		front	在所有其他标签页/窗口之前放置新的目的地标签页/窗口
		back	在所有其他标签页/窗口之后放置新的目的地标签页/窗口

11. Marquee 属性

属性	描述
marquee-direction	设置移动内容的方向
marquee-play-count	设置内容移动的次数
marquee-speed	设置内容滚动得多快
marquee-style	设置移动内容的样式

12. 多列属性（Multi-column）

属性	描述	值	属性值说明
column-count	规定元素应该被划分的列数	number	元素内容将被划分的最佳列数
		auto	由其他属性决定列数，如“column-width”
column-fill	规定如何填充列（是否进行协调）	balance	对列进行协调。浏览器应对列长度的差异进行最小化处理
		auto	按顺序对列进行填充，列长度会各有不同

续表

属性	描述	值	属性值说明
column-gap	规定列之间的间隔	length	把列间的间隔设置为指定的长度
		normal	规定列间间隔为一个常规的间隔，建议值是 1em
column-rule	设置所有 column-rule-* 属性的简写属性	column-rule-width	设置列之间的宽度规则
		column-rule-style	设置列之间的样式规则
		column-rule-color	设置列之间的颜色规则
column-rule-color	规定列之间的颜色规则	red，green，blue	预定义的颜色值
		#FF0000，#FF6600	十六进制颜色值，也是最常用的定义颜色的方式
		rgb(255,0,0)或 rgba(255,0,0，1)	rgb 代码，参数代表红、绿、蓝三色的数值，a 代表透明度
column-rule-style	规定列之间规则的样式	none	定义没有规则
		hidden	定义隐藏规则
		dotted	定义点状规则
		dashed	定义虚线规则
		solid	定义实线规则
		double	定义双线规则
		groove	定义 3D grooved 规则。该效果取决于宽度和颜色值
		ridge	定义 3D ridged 规则。该效果取决于宽度和颜色值
		inset	定义 3D inset 规则。该效果取决于宽度和颜色值
		outset	定义 3D outset 规则。该效果取决于宽度和颜色值
column-rule-width	规定列之间的宽度规则	thin	定义纤细规则
		medium	定义中等规则
		thick	定义宽厚规则
		length	规定规则的宽度
column-span	规定元素应该横跨的列数	1	元素应横跨一列
		all	元素应横跨所有列
column-width	规定列的宽度	auto	由浏览器决定列宽
		length	规定列的宽度
columns	规定设置 column-width 和 column-count 的简写属性	column-width	列的宽度
		column-count	列数

13. Paged Media 属性

属性	描述
fit	示意如何对 width 和 height 属性均不是 auto 的被替换元素进行缩放
fit-position	定义盒内对象的对齐方式
image-orientation	规定用户代理应用于图像的顺时针方向旋转
page	规定元素应该被显示的页面的特定类型
size	规定页面内容包含框的尺寸和方向

14. CSS 文本属性（Text）

属性	描述	值	属性值说明
hanging-punctuation	规定把标点符号放在文本整行的开头还是结尾的行框之外	none	不在文本整行的开头和结尾的行框之外放置标签符号
		first	标点附着在首行开始边缘之外
		last	标点附着在首行结尾边缘之外
		allow-end	如果没有其他设置，在行结束时的一个停止或逗号挂起
		force-end	行结束时的停止或逗号挂起
punctuation-trim	规定是否对标点字符进行修剪	none	不修剪开启或闭合标点符号
		start	修剪每行结尾的开启标点符号
		end	修剪每行结尾的闭合标点符号
		allow-end	如果另有不适合之前的理由，修剪每行末尾的结束标点符号
		adjacent	根据相邻字符修剪开口 punctuation 修剪
text-emphasis	向元素的文本应用重点标记以及重点标记的前景色	text-emphasis-style	向元素的文本应用重点标记
		text-emphasis-color	定义重点标记的前景色
text-justify	规定当 text-align 设置为"justify"时的对齐方法	auto	浏览器决定齐行算法
		none	禁用齐行
		inter-word	增加/减少单词间的间隔
		Inter-ideograph	用表意文本来排齐内容
		inter-cluster	只对不包含内部单词间隔的内容进行排齐
		distribute	类似报纸版面，除了在东亚语系中最后一行是不齐行的
		kashida	通过拉伸字符来排齐内容
text-outline	规定文本的轮廓	thickness	必需，轮廓的粗细
		blur	可选，轮廓的模糊半径
		color	必需，轮廓的颜色
text-overflow	规定当文本溢出包含元素时发生的事情	clip	修剪文本
		ellipsis	显示省略符号来代表被修剪的文本
		string	使用给定的字符串来代表被修剪的文本
text-shadow	向文本添加阴影	h-shadow	必需，水平阴影的位置，允许负值
		v-shadow	必需，垂直阴影的位置，允许负值

续表

属性	描述	值	属性值说明
text-shadow	向文本添加阴影	blur	可选，模糊的距离
		color	可选，阴影的颜色
text-wrap	规定文本的换行规则	normal	只在允许的换行点进行换行
		none	不换行，元素无法容纳的文本会溢出
		unrestricted	在任意两个字符间换行
		suppress	压缩元素中的换行。浏览器只在行中没有其他有效换行点时进行换行
word-break	可以让浏览器实现在任意位置的换行	normal	使用浏览器默认的换行规则
		break-all	允许在单词内换行
		keep-all	只能在半角空格或连字符处换行
word-wrap	允许对长的不可分割的单词进行分割并换行到下一行	normal	只在允许的断字点换行（浏览器保持默认处理）
		break-word	在长单词或 URL 地址内部进行换行
text-align-last	设置如何对齐最后一行或紧挨着强制换行符之前的行		

15. 2D/3D 转换属性（Transform）

属性	描述	值	属性值说明
transform	向元素应用2D或3D 转换	none	定义不进行转换
		matrix (n,n,n,n,n,n)	定义 2D 转换，使用 6 个值的矩阵
		matrix3d (n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n)	定 3D 转换，使用 16 个值的 4×4 矩阵
		translate(x,y)	定义 2D 转换
		translate3d (x,y,z)	定义 3D 转换
		translateX(x)	定义转换，只是用 X 轴的值
		translateY(y)	定义转换，只是用 Y 轴的值
		translateZ(z)	定义 3D 转换，只是用 Z 轴的值
		scale(x,y)	定义 2D 缩放转换
		scale3d(x,y,z)	定义 3D 缩放转换
		scaleX(x)	通过设置 X 轴的值来定义缩放转换
		scaleY(y)	通过设置 Y 轴的值来定义缩放转换
		scaleZ(z)	通过设置 Z 轴的值来定义 3D 缩放转换
		rotate(angle)	定义 2D 旋转，在参数中规定角度
		rotate3d (x,y,z,angle)	定义 3D 旋转
		rotateX(angle)	定义沿着 X 轴的 3D 旋转
		rotateY(angle)	定义沿着 Y 轴的 3D 旋转
		rotateZ(angle)	定义沿着 Z 轴的 3D 旋转
		skew (x-angle,y-angle)	定义沿着 X 和 Y 轴的 2D 倾斜转换

续表

属性	描述	值	属性值说明
transform	向元素应用2D或3D转换	skewX(angle)	定义沿着 X 轴的 2D 倾斜转换
		skewY(angle)	定义沿着 Y 轴的 2D 倾斜转换
		perspective(n)	为 3D 转换元素定义透视视图
transform-origin	允许改变被转换元素的位置	x-axis	定义视图被置于 X 轴的何处，可能的值为 left|center|right|length|%
		y-axis	定义视图被置于 Y 轴的何处，可能的值为 top|center|bottom|length|%
		z-axis	定义视图被置于 Z 轴的何处，可能的值为 length
transform-style	规定被嵌套元素如何在 3D 空间中显示	flat	子元素将不保留其 3D 位置
		preserve-3d	子元素将保留其 3D 位置
perspective	规定 3D 元素的透视效果	number	元素距离视图的距离，以像素计
		none	默认值，与 0 相同，不设置透视
perspective-origin	规定 3D 元素的底部位置	x-axis	定义该视图在 X 轴上的位置，默认值为 50%，可能的值为 left|center|right|length|%
		y-axis	定义该视图在 Y 轴上的位置，默认值为 50%，可能的值为 top|center|bottom|length|%
backface-visibility	定义元素在不面对屏幕时是否可见	visible	背面是可见的
		hidden	背面是不可见的

16. 过渡属性（Transition）

属性	描述	值	属性值说明
transition	简写属性，用于在一个属性中设置 4 个过渡属性	transition-property	规定设置过渡效果的 CSS 属性的名称
		transition-duration	规定完成过渡效果需要多少秒或毫秒
		transition-timing-function	规定速度效果的速度曲线
		transition-delay	定义过渡效果何时开始
transition-property	规定应用过渡的 CSS 属性的名称	none	没有属性会获得过渡效果
		all	所有属性都将获得过渡效果
		property	定义应用过渡效果的 CSS 属性名称列表，列表以逗号分隔
transition-duration	定义过渡效果花费的时间	time	规定完成过渡效果需要花费的时间（以秒或毫秒计），默认值是 0，意味着不会有效果
transition-timing-function	规定过渡效果的速度曲线	linear	规定以相同速度开始至结束的过渡效果（等 cubic-bezier(0,0,1,1)）
		ease	规定慢速开始，然后变快，然后慢速结束的过渡效果（cubic-bezier(0.25,0.1,0.25,1)）

续表

属性	描述	值	属性值说明
transition-timing-function	规定过渡效果的速度曲线	ease-in	规定以慢速开始的过渡效果（等于 cubic-bezier(0.42,0,1,1)）
		ease-out	规定以慢速结束的过渡效果（等于 cubic-bezier(0,0,0.58,1)）
		ease-in-out	规定以慢速开始和结束的过渡效果（等于 cubic-bezier(0.42,0,0.58,1)）
		cubic-bezier(n,n,n,n)	在 cubic-bezier 函数中定义自己的值，可能的值是 0～1 的数值
transition-delay	规定过渡效果何时开始	time	规定在过渡效果开始之前需要等待的时间，以秒或毫秒计

17. 用户界面属性（User-interface）

属性	描述	值	属性值说明
appearance	允许用户将元素设置为标准用户界面元素的外观	normal	将元素呈现为常规元素
		icon	将元素呈现为图标（小图片）
		window	将元素呈现为视口
		button	将元素呈现为按钮
		menu	将元素呈现为一套供用户选择的选项
		field	将元素呈现为输入字段
box-sizing	允许用户以确切的方式定义适应某个区域的具体内容	content-box	宽度和高度分别应用到元素的内容框。在宽度和高度之外绘制元素的内边距和边框
		border-box	为元素设定的宽度和高度决定了元素的边框盒。也就是说，为元素指定的任何内边距和边框都将在已设定的宽度和高度内进行绘制。通过由已设定的宽度和高度分别减去边框和内边距才能得到内容的宽度和高度
		inherit	规定应从父元素继承 box-sizing 属性的值
icon	为创作者提供使用图标化等价物来设置元素样式的能力	auto	使用由浏览器提供的默认通用图标
		URL	引用列表中的一个或多个图标，列表用逗号分隔
		inherit	规定应从元素继承 icon 属性的值
nav-down	规定在使用 arrow-down 导航键时向何处导航	auto	浏览器决定导航到哪个元素
		id	规定被导航元素的 id
		target-name	规定被导航的目标框架
		inherit	规定应从父元素继承 nav-down 属性的值
nav-index	设置元素的 tab 键的控制次序	auto	由浏览器分配元素的导航键的控制次序
		number	指示元素的导航键的控制次序，1 代表第一个
		inherit	规定应从父元素继承 nav-index 属性的值

续表

属性	描述	值	属性值说明
nav-left	规定在使用 arrow-left 导航键时向何处导航	auto	浏览器决定导航到哪个元素
		id	规定被导航元素的 id
		target-name	规定被导航的目标框架
		inherit	规定应从父元素继承 nav-left 属性的值
nav-right	规定在使用 arrow-right 导航键时向何处导航	auto	浏览器决定导航到哪个元素
		id	规定被导航元素的 id
		target-name	规定被导航的目标框架
		inherit	规定应从父元素继承 nav-right 属性的值
nav-up	规定在使用 arrow-up 导航键时向何处导航	auto	浏览器决定导航到哪个元素
		id	规定被导航元素的 id
		target-name	规定被导航的目标框架
		inherit	规定应从父元素继承 nav-up 属性的值
outline-offset	对轮廓进行偏移，并在超出边框边缘的位置绘制轮廓	length	轮廓与边框边缘的距离
		inherit	规定应从父元素继承 outline-offset 属性的值
resize	规定是否可由用户对元素的尺寸进行调整	none	用户无法调整元素的尺寸
		both	用户可调整元素的高度和宽度
		horizontal	用户可调整元素的宽度
		vertical	用户可调整元素的高度

Responsive Web Design

Appendix

3

附录 3

CSS3 中需要加浏览器私有前缀的属性

1. border-radius

属性名	
border-radius	border-bottom-right-radius
border-top-left-radius	border-bottom-left-radius
border-top-right-radius	

2. box-shadow

3. animation

属性名	
animation	animation-fill-mode
animation-name	animation-iteration-count
animation-duration	animation-play-state
animation-delay	animation-timing-function
animation-direction	@keyframes

4. transition

属性名	
transition	transition-delay
transition-property	transition-timing-function
transition-duration	

5. transform

属性名	
transform	perspective-origin
transform-origin	transform-style
perspective	backface-visibility

6. gradient

属性名	
linear-gradient()	repeating-linear-gradient()
radial-gradient()	repeating-radial-gradient()

7. box-sizing

8. filter

9. columns

属性名	
columns	column-rule-style
column-width	column-span
column-gap	column-fill
column-rule	break-before

续表

属性名	
column-rule-color	break-after
column-rule-width	break-inside
column-count	

10. user-select

11. flex

属性名	
display	flex-flow
flex	justify-content
flex-grow	order
flex-shrink	align-items
flex-basis	align-self
flex-direction	align-content
flex-wrap	

12. calc()

13. background

属性名	
background-clip	background-size
background-origin	

14. font-feature

属性名	
font-feature-settings	font-language-override
font-variant-ligatures	font-kerning

15. border-image

16. ::selection

17. ::placeholder

18. hyphens

19. ::fullscreen

20. tab-size

21. Intrinsic & extrinsic sizing

属性名	
max-content	fit-content
min-content	

22. cursor

属性名	
zoom-in	grab
zoom-out	grabbing

23. position

24. position:sticky

25. touch-action

26. text-decoration

属性名	
text-decoration-style	text-decoration-color
text-decoration-line	

27. text-size-adjust

28. mask

属性名	
clip-path	mask-origin
mask	mask-position
mask-clip	mask-repeat
mask-composite	mask-size
mask-image	

29. box-decoration-break

30. shapes

属性名	
shape-outside	shape-margin
shape-image-threshold	

31. object-fit

32. object-position

33. resolution

属性名	
min-resolution	max-resolution

34. @viewport

35. text-emphasis

36. text-align-last

37. text-overflow

38. CSS Logical Properties

属性名	
margin-block-start	padding-inline-start
margin-block-end	padding-inline-end
margin-inline-start	border-block-start
margin-inline-end	border-block-end
padding-block-start	border-inline-start
padding-block-end	border-inline-end

39. image-rendering

40. image-rendering: pixelated

41. CSS Images

属性名	
cross-fade()	element()
image-set()	

42. CSS Masks

属性名	
mask-border	mask-border-width
mask-border-source	mask-border-outset
mask-border-slice	mask-border-repeat

43. CSS Filter

属性名	
filter()	backdrop-filter

44. CSS Scroll Snap Points

属性名	
scroll-snap-type	scroll-snap-points-x
scroll-snap-coordinate	scroll-snap-points-y
scroll-snap-destination	

45. CSS Regions

属性名	
flow-into	region-fragment
flow-from	

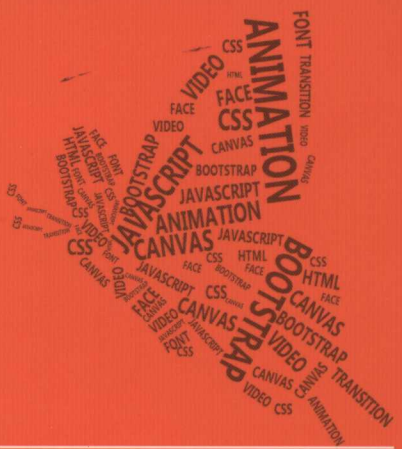
46. CSS Writing Modes

属性名	
writing-mode: horizontal-tb	writing-mode: vertical-lr
writing-mode: vertical-rl	

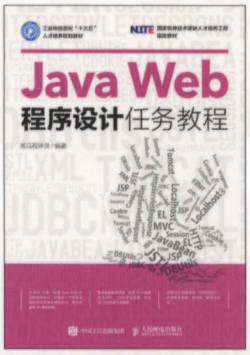
47. ::backdrop
48. ::read-only
49. cursor: grab
50. cursor: grabbing
51. Intrinsic & extrinsic sizing
52. fill
53. text-emphasis-position

加微信：208695827
QQ：208695827

快来领取！



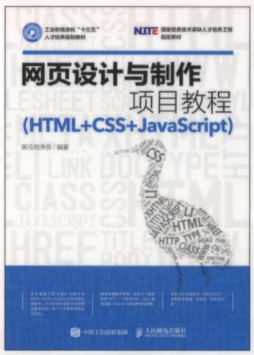
应用型 IT 套系教材推荐 RECOMMEND APPLICATION IT SERIES TEXTBOOK



978-7-115-43936-9

978-7-115-43937-6

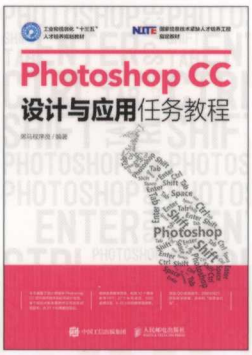
978-7-115-43934-5



978-7-115-43935-2

978-7-115-43938-3

978-7-115-43933-8



978-7-115-43932-1

免/费/提/供
PPT等教学相关资料

人邮教育
www.ryjiaoyu.com

教材服务热线：010-81055256
反馈 / 投稿 / 推荐信箱：315@ptpress.com.cn
人民邮电出版社教育服务与资源下载社区：www.ryjiaoyu.com

封面设计：董志桢

ISBN 978-7-115-43934-5

ISBN 978-7-115-43934-5
定价:42.00 元